MASTER IN COMPUTER SCIENCE

AODV-FUUREX

Contribution to a large scale validation of its implementation

Eugène, Th

Award date:
2012

Awarding institution:
University of Namur

University of Namur

Faculty of Computer Science

Master thesis

AODV-FUUREX : Contribution to a large scale validation of its implementation

Thomas Eugène

Dissertation submitted in partial fulfilment for the Master's degree in Computer Science

University of Namur

Faculty of Computer Science

Master thesis

AODV-FUUREX : Contribution to a large scale validation of its implementation

Thomas Eugène

Supervisors
Laurent Schumacher - Roberto Canonico

August 31, 2012

Dissertation submitted in partial fulfilment for the Master's degree in Computer Science

Abstract

Wireless technology are very widespread and used for a lot of applications. These wireless networks are generally in "infrastructure mode" and do not use the other possibility provided by wireless technology. Wireless mesh network usages are really scarce.

This document focus on AODV-FUUREX implementation, a secure version of the routing algorithms AODV. This implementation is compared with a basic version (AODV-UU).

Experiment scripts are written to work with OMF, a framework for managing experimental platform. Different malicious behaviours are simulated to test AODV-FUUREX.

These tests are executed on two different environments. The first one is a virtual testbed which was used to develop the experiment scripts and check that all the elements work good.
The second one is a real environment which is used with bigger topologies.

Furthermore, the implementation is improved.

keywords : AODV, Wireless Mesh Network (WMN), OMF

Résumé

Les technologies sans-fils sont très répandues et utilisées pour un grand nombre d'applications. Ces réseaux sans-fils sont généralement exploités en "mode infrastructure" mais n'exploitent pas les autres possibilités offertes par les technologies sans-fils. L'utilisation des réseaux sans-fils en réseau maillé reste marginale.

Ce document se concentre sur l'implémentation d'AODV-FUUREX, une version sécurisée de l'algorithme de routage AODV. Cette implémentation est comparée à une version de base (AODV-UU).

Les scripts d'expérimentations sont écrits pour fonctionner sur OMF, une plateforme de gestion d'environnement de test. Différents comportements malicieux sont simulés afin de tester le bon fonctionnement d'AODV-FUUREX.

Ces tests sont effectués sur deux environnements différents. Le premier est un environnement virtuel qui a permis de développer les scripts d'expérimentations et de tester le bon fonctionnement des différents éléments.
Le deuxième est un environnement réel qui est utilisé avec de plus grandes topologies.

En plus de cette analyse, certaines améliorations sont apportées à l'implémentation.

Mots-clés : AODV, réseau maillé sans-fils, OMF

Acknowledgements

This dissertation would not have been possible without the help and the support of several people who contribute to this completion.

I would like to thank:

Professor Schumacher for his availability, and for all the advice, support and corrections that I receive throughout this year.

Professor Roberto Canonico for the welcome, the supervision of my internship and the help that he gave me during all my stay at Naples.

Giovanni Di Stasi for the time that he takes to answer to my questions and help me, the rewarding discussions and for his works on OMF which was really helpful.

Professor Giorgio Ventre and the COMICS laboratory researchers for their welcoming during my internship.

My family for their support during this year and all my studies.

Thibault for his advice, his help and his support.

All my friends and people who support me during my studies.

Contents

Acronyms

- AODV: Ad-hoc On Demand Vector
- AODV-UU: Ad-hoc On Demand Vector - Uppsala University
- AODV-FUUREX: Ad-hoc On Demand Vector - Fundp Uppsala University Repudiation EXtension
- CPU: Central Processing Unit
- D-ITG: Distributed Internet Traffic Generator
- DSDV: Destination-Sequenced Distance Vector
- DSR: Dynamic Source Routing
- HTTP: HyperText Transport Protocol
- IP: Internet Protocol
- MANET: Mobile Ad-hoc Network
- OLSR: Optimized Link State Routing
- OMF: cOntrol and Management Framework
- REFACING:RElationship-Familiarity-Confidence-INteGrity
- RERR: Route ERRor
- RREP: Route REPly
- RREQ: Route REQuest
- TCP: Transmission Control Protocol
- TORA: Temporally-Ordered Routing Algorithm
- TTL: Time To Live
- UDP: User Datagram Protocol
- WMN: Wireless Mesh Network
- ZRP: Zone Routing Protocol

Chapter 1

Introduction

Internet is a wonderful tool which helps people to communicate and collaborate without a problem of distance.
Internet was now became mandatory to communicate with people, to exchange or get information, to manage bank account, to play, ... Internet usages are endless.

Internet is more and more present in our lives. At home, it is no more limited to a computer on a desk. Internet must be available in each room. To permit it easily wireless networks are commonly used. Internet provider offer often wireless access points with their Internet access.

More and more devices become able to communicate through wireless technology. It was computer first, it's phone and tablet now and it will be a lot of other devices in the future. Wireless Technology is everywhere.

Actual wireless networks are centralized and have a limited range. To use Internet in the street the only solution is to use a mobile data network which is not free of charge.
With the deployment of wireless technology, it becomes imaginable to create a wireless device network. This network where all the devices are interconnected creates a huge web covering cities or countries.

A big uncentralized network could bring new perspectives.
This type of technology could be a good solution by avoiding censorship and permitting data exchange in less developed countries. Another big advantage of this type of network is to be able to roam using the network.

A lot of research has been made on this type of technology. Some projects are well developed and propose a working implementation. But this technology is not generally available and is not used everyday.
Basics implementations of this technology are often unsecure.

One example of this technology is called AODV[3]; this thesis will focus on a secure implementation of this network technology.

The secure version is tested and compared with a basic version. Test are made in two different testbeds. The first testbed is a virtual environment which has the biggest advantage to be all the

time available. The problem is that the results obtains in this type of environment hide parameters like radio interferences, distance between nodes,...
A testbed with real nodes is used to check others parameters.

To test the security and compare the behaviour of the two implementations, malicious behaviour are launched during the experiments.

The goals of theses tests are to check the robustness, the scalability, the ability to detect malicious behaviour and to analyze the strengths and weakness.

The results of these experiments are discussed and analysed to propose improvements and solution to detected problems.

Chapter 2

State of Art

2.1 Wireless Network

A wireless network is a computer network without cables. The nodes (devices with wireless connection) are connected using radio waves. There are different types of standard wireless connection like IEEE 802.11 (Wi-Fi), IEEE 802.15.4 which are the most common.

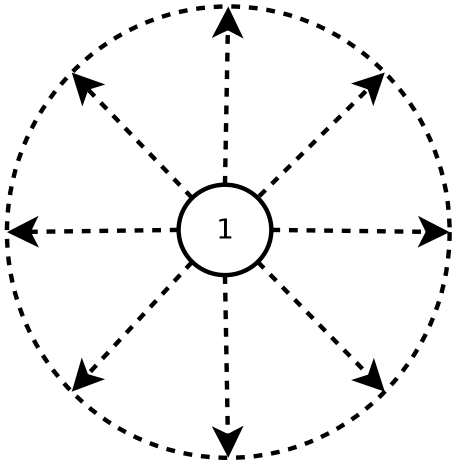


Figure 2.1: Wireless transmission

For a normal node without directive antenna, the waves are sent all around the node (Omnidirectional radiation pattern) like on Figure 2.1.
Wireless nodes can be connected in two different modes:

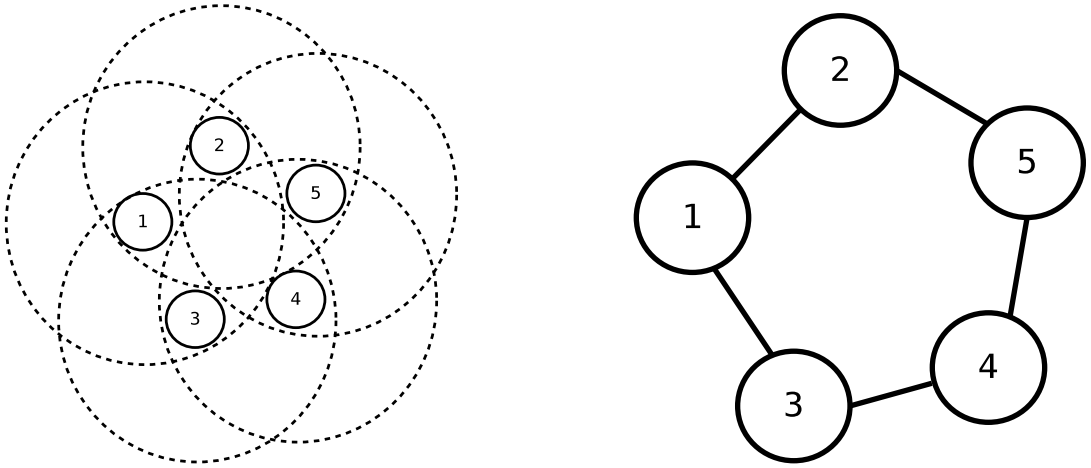
- **Infrastructure mode:** In this case all the devices are managed by one device which is called the access point. All the connections between nodes are relayed by this central node. To be in the network, all the nodes must be in range of the access point. If the access point is down all the network is unreachable.
  This mode is commonly used by home user as access point for Internet.

(a) Within radio range in dotted line. (b) Simplified model

Figure 2.2: Wireless network representation

Figure 2.2a shows an infrastructure wireless network with a master node represented by the "M". Only the nodes in the range of the master node are in the network.
A simplified model is represented on the Figure 2.2b.

- **Ad-hoc mode:** This mode allows connection between nodes without access point. All devices on the same network are directly connected at the reachable neighbours.



(a) Within radio range in dotted line. (b) Simplified model

Figure 2.3: Wireless network representation

On Figure 2.3a, the wireless ad-hoc network is represented with dotted lines circles which are the range of the nodes. To make it easier the radio waves circles are replaced by lines between nodes which show the communication between nodes like on Figure 2.3b. If there is no line between nodes, the nodes are not connected.
All the network in this document will be represented with the simplified model which is clearer than the other.

2.2 Wireless Mesh Network (WMN)

A WMN is a special type of ad-hoc connection where all the nodes are interconnected in a mesh topology. There is no central node. The coverage of the network depends of the connected devices.

Each connected node increases the coverage.
To join another device on the network, the signal can be transmitted through intermediate nodes to reach the receiver. Each node is a router in the mesh.
Some applications of wireless network are presented in [8] and [11]. There are diverse applications like broadband Internet access, WLAN coverage and mobility, Mobile data network support, ... All these applications are still in development or in pilot projects. WMN are really cost-effective. The nodes build the network themselves. There is no cable needed.
The redundancy of the links is a big advantage. When a link is broken, another route can be used to transfer data. To make it possible special routing protocols are needed.

2.3 Mobile Ad-hoc Network (MANET)

A MANET is a WMN for mobile device network. In this type of network all the nodes are free to move. The links and the topology of the network are changing all the time. Despite these changes, the network must maintain the information required to route the data.

2.4 Routing Protocols

To manage and route the data through ad-hoc wireless networks special routing protocols have been implemented. There are three main types of protocols.

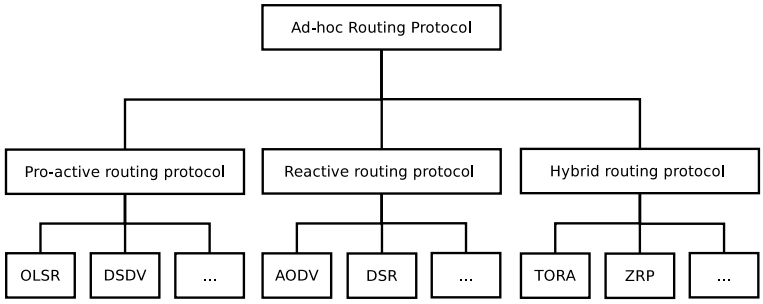


Figure 2.4: Routing protocols

2.4.1 Pro-active routing protocols

The pro-active routing protocols need to know all the nodes and all the paths of the network before sending data. A table of routes is maintained on each node with the possible route to reach each other. This type of protocol needs a lot of data transfer to keep all the routes up-to-date. Another drawback is the time needed to recovery from a failure.

2.4.2 Reactive routing protocols

Reactive routing is also called on-demand routing protocol. These protocols are looking for a route when it is needed. A route request message is broadcasted by the sender to find the best route to the receiver. This request takes time, there is latency to find route. All these broadcast messages could clog the network.

2.4.3 Hybrid routing protocols

This type of routing protocol takes advantage of the pro-active and reactive routing protocol. In this protocol, some nodes are connected with a pro-active routing protocol. These nodes constitute a skeleton of the network. From these main nodes, the routes to the destinations are created by a reactive protocol.

2.5 Secure ad-hoc routing protocols

Wireless network security threats are more important than wired networks. The biggest problem is due to the air communications. In ad-hoc network another big threat is that the nodes must rely on others.
These weakness make ad-hoc network really sensitive to attack and malicious behaviour.

2.5.1 Malicious Node

A complete list of attack is presented in [9]. Only the most common are presented here.

- Black hole attack: this attack is also called packet drop attack. The black hole node drops the packets which must be relayed to other nodes.
- Grey hole attack: this type of attack is like the Black hole but with a selection of the dropped packets. For example, the node drops only the HTTP packets.
- Wormhole attack: Wormhole attack are composed of 2 nodes. One node records the received traffic and sends it by another network to another node located elsewhere in the topology which replays the traffic recorded. In this attack, the data can also be manipulated by the malicious nodes.
- Flooding: By this attack a node could make a denial of service on the network. If more than one node are controlled to flood, this attack is more powerful.

2.5.2 Security solutions

The security of ad-hoc network is a research subject which is cover by articles like [10], [2] and [12]. The current proposals for solutions exploit two different ways:

- Cryptographic key usage: the usage of keys can authenticate the node and avoid problems with malicious nodes. But the biggest problem with this method is the deployment of the keys. The key must be shared before deployment or by a trusted intermediate. Both solutions are a problem for incremental network composition.
  Another problem is that cryptographic keys needs computation power and energy that must be saved as much as possible for mobile devices.
- Reputation base: In this case, for each packet the node checks if the packet is forwarded by the next-hop using the ability to listen the traffic of its neighbour. The node has a table with the reputation of their neighbour. According to this reputation, the most trusted path is selected to transfer data.

2.6 AODV routing protocol

AODV is a reactive routing protocol for wireless ad-hoc network. This routing protocol is defined by the IETF in the RFC 3561 [3].
This protocol is defined to be adaptive to the changes of topology. One of the aims of this protocol is to be suitable for MANETS.
As reactive routing protocol, AODV is creating routes when they are needed for data transfer.
The route information are not stored in the packet. Each node has received the information to choose the best next-hop to reach the destination. There are four message types in AODV which are transmitted over UDP.

2.6.1 Route Request (RREQ)

Message sent by the source to discover a path to the destination when there is none known yet.

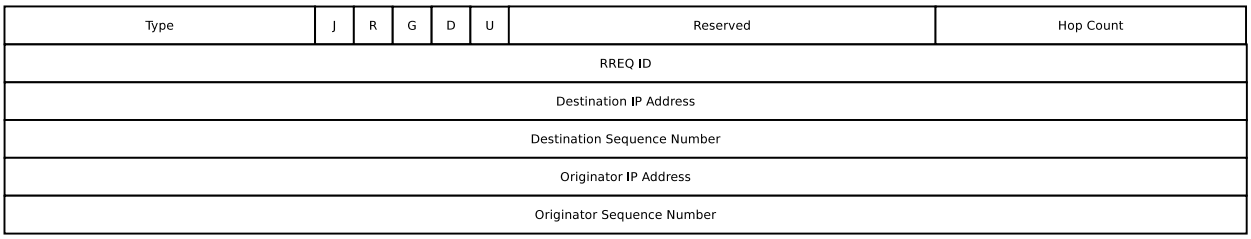


Figure 2.5: RREQ packet representation

- Type is set at the value 1.
- **J**oin Flag: is used to join a multicast group.
- **R**epair Flag: is used to repair multicast route.
- **G**ratuitous Flag: if a intermediate node creates a RREP to the originator a RREP must be sent to the destination.
- **D**estination only Flag: the RREQ must be forwarded by the intermediate nodes to the destination. Only the destination can generate a RREP for this RREQ
- **U**nknown sequence number flag: indicates that the sequence number is unknown. Used whenever a route has been created to join the destination.
- Reserved: is ignored
- Hop Count: number of node the RREQ went through.
- RREQ ID: unique value used to identify the RREQ.
- Destination IP Address: Address of the receiver.
- Destination Sequence Number: value used to identify the route to the destination and to check the freshness of the route. Higher value = fresher route.
- Originator IP Address: Address of the sender.
- Originator Sequence Number: value used to identify the route to the originator and to the check the freshness of the route. Higher value = fresher route.

2.6.2 Route Reply (RREP)

Message in reply of the RREQ from the destination to the source. This message provides each intermediate node information about the best path to reach destination.
This type of message is also used as Hello message. Hello messages are used by nodes to signal their presence at their neighbours. To reach only the neighbours the TTL is set to 1 for the Hello message.

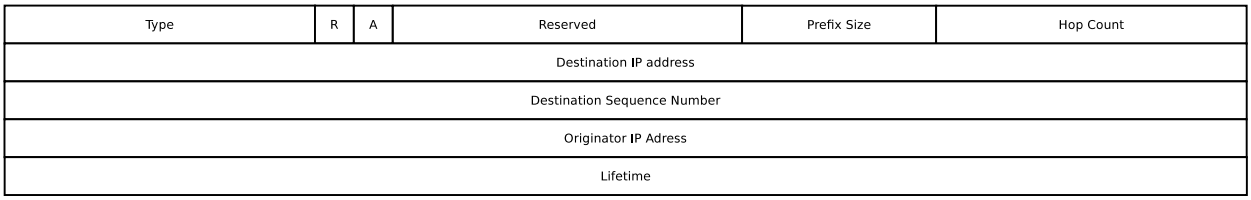


Figure 2.6: RREP packet representation

- Type is set at the value 2.
- **R**epair Flag: is used to repair multicast route.
- **A**cknowledgement required Flag: is used to ask an acknowledgement if there is suspicion of unidirectional link.
- Reserved: is ignored
- Prefix Size: used for supplied route in a specific subnet.
- Hop Count: copy of the RREQ hop count.
- Destination IP Address: the IP address of the destination specified in the RREQ.
- Destination Sequence Number: value used to identify the route to the destination and to check the freshness of the route. Higher value = fresher route.
- Originator IP Address: address of the originator of the RREQ is also the destination of the RREP.
- Originator Sequence Number: value used to identify the route to the originator and to the check the freshness of the route. Higher value = fresher route.

2.6.3 Route Reply Acknowledgement (RREP-ACK)

Message sent as Acknowledgement of RREP. This type of message is used when there is a risk of unidirectional links.

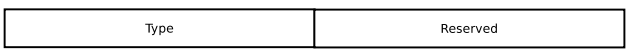


Figure 2.7: RREP-ACK packet representation

- Type is set at the value 4.
- Reserved field is ignored.

2.6.4 Route Error (RERR)

Message used to advertise a problem on the route.

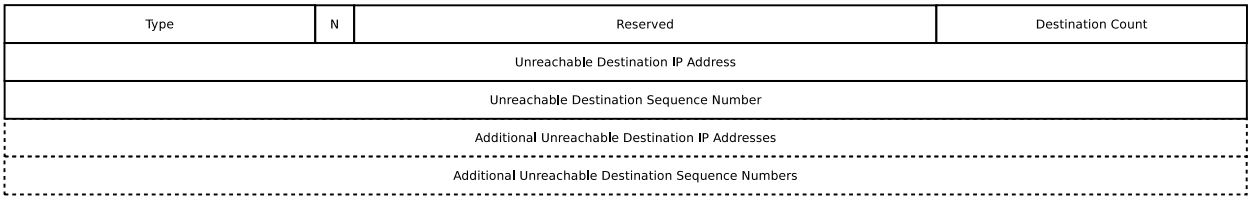


Figure 2.8: RERR packet representation

- Type is set at the value 3.
- No delete flag: is used to signal that the route is repaired and that there is no need to delete it.
- Reserved field: is ignored
- Destination count: is the number of unreachable destination(s) due to a link break. This value will add lines to the packet (minimum value is 1).
- Unreachable Destination IP Address: is the IP of the unreachable node.
- Unreachable Destination Sequence Number: is the ID of the broken routes.
- The others fields are created to add more unreachable destinations. For each pair of field, the destination count must be incremented.

2.6.5 Routing table

Each node maintains a routing table with all the information needed to route data. In this routing table we can find:

- Destination IP Address
- Destination Sequence Number which is used to identify the most recent route.
- Valid Destination Sequence Number flag
- Routing flags (valid, invalid, repairable, being repaired)
- Network Interface
- Hop Count (number of hops needed to reach destination)
- Next Hop to reach the destination
- Lifetime of the route until expiration.

2.7 AODV in use

The best way to understand how AODV works is by an example.

2.7.1 Creation of the network

When a node arrives on the network, it broadcasts Hello messages to signal its presence. The neighbours which receive the message add the node in their routing table.

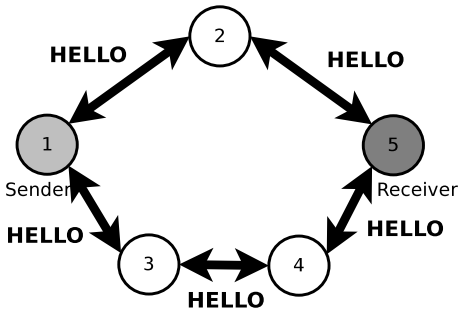
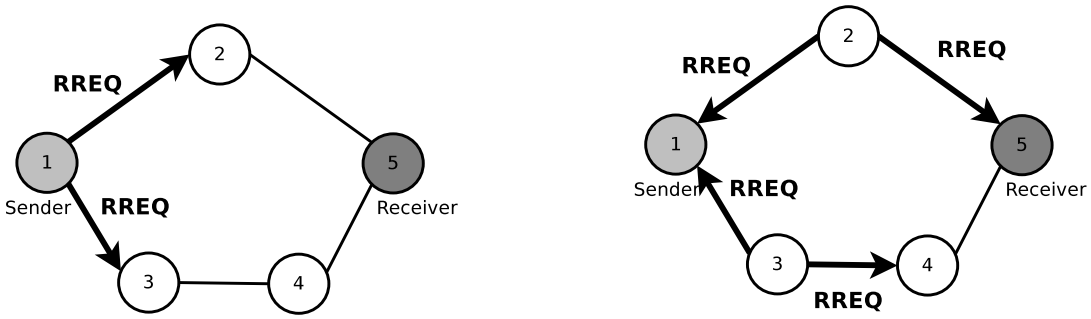


Figure 2.9: Hello message broadcast

Hello message or other AODV control messages must be sent regularly to keep links alive with the neighbours. Without Hello message or other AODV control message from a node the neighbour drops the route and the node is no longer directly available.

2.7.2 Route creation

In this example, Node 1 is the sender and Node 5 is the receiver. It is the first data transfer in the network and there is no route created before.

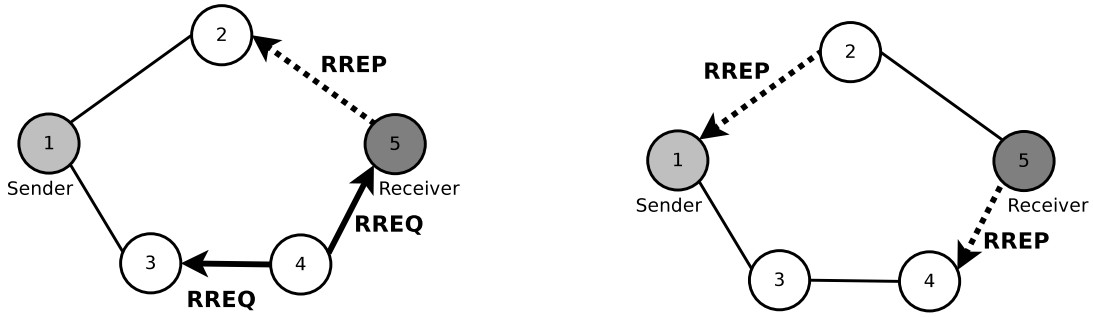


(a) Sender send RREQ (b) RREQ propagation

Figure 2.10: RREQ process

1. The sender wants to send data to the receiver and there is no route for the destination in the routing table. The node broadcasts RREQ to its neighbours.
   The RREQ is created with the needed data.
   In the Destination Sequence Number field, the identifier of the last route to the destination in the routing table. When it is the first RREQ to this destination a flag is activated.
   The originator sequence number is incremented and added to the RREQ like the RREQ ID which is also incremented by one. Each node manages is own RREQ ID counter.
   The RREQ ID and the originator IP are saved by the node to avoid the reprocessing of the RREQ received from its neighbours.
   The Hop Count is set to 0. The RREQ is sent to all the neighbours like on Figure 2.10a.

2. The RREQ is processed by Node 2 and Node 3. With this RREQ the route to the previous hop is updated. The originator IP and the RREQ ID are checked and the packet is dropped if this packet was received before.
   Other RREQ are treated by the node. The hop count is increased by one and the reverse route is created or updated to be ready to send RREP if needed.
   If there is no route to the destination in the routing table, RREQs are broadcasted to the neighbours like we can see on Figure 2.10b.

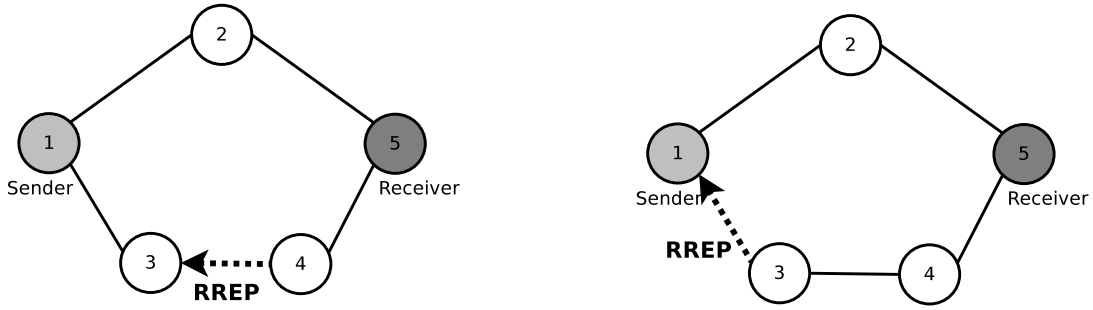


(a) RREQ propagation and RREP for route 1 (b) RREP for route 1 and 2

Figure 2.11: RREQ and RREP process

3. The sender drops the RREQ received from Node 2 and Node 3 which are identified by the originator IP and the RREQ ID. In the same time node 4 process the RREQ like Node 2 and Node 3 at the previous step. The RREQ received by the receiver from Node 2 is processed and a RREP is generated. The sequence number is incremented if the packet sequence number is the same that the node's one and added at the RREP. The originator IP and the originator sequence number are taken in the RREQ and inserted in the RREP. The RREP is unicasted to the next hop following the information added in the routing table. (Figure 2.11a).

4. At this step the 2 RREQs have reached the receiver. The RREQ from Node 4 is processed like the request from Node 2.
   The RREQ from Node 4 is discarded by Node 3.
   Node 2 received the RREP from the receiver and changed the sequence number with the sequence number recorded at the processing of the RREQ on Figure 2.10a. The routing table entry is updated with the information of the RREP and update the next hop for the destination from the originator (Figure 2.11b).



(a) RREP for route 2 (b) RREP for route 2

Figure 2.12: RREP process

5. The sender has a path to send data to the destination. The route from the RREP is updated in the routing table and the data can be sent to the receiver through the established path. In the same time, the RREP on the second path is processed, like in the previous step for Node 2, by Node 4 and forwarded to Node 3 like on Figure 2.12a.

6. The second RREP which arrived at the sender is dropped. AODV keeps only the shortest path. The others are dropped.

2.7.3 Unidirectional link

If there is a unidirectional link on the shortest path to the destination, the originator will never receive the RREP. To avoid this problem, when a originator haven't response from the destination a new RREQ is sent after a timeout.
When a node detects that the RREP transmission have failed, this node is blacklisted during a fixed time. All the RREQs from this node will be dropped to let the chance to the same RREQ from another path to be processed. To be sure that the route is created the destination or a node forwarding the RREP can ask an acknowledgement at the sender by adding a flag 'A' in the packet.

2.7.4 Broken link

When there is a route problem, the route must be invalidated and the affected nodes warned by RERR message. RERR are sent by node if:

- the link to a next hop node of an active route (valid route in the routing table) is broken
- it received data for a destination which is not in its routing table.
- it received a RERR from a neighbour.

In this case, the node checks all the destination in its routing table and sends RERR to warn the other nodes. The RERR is generally broadcasted with a TTL set to 1 to warn only neighbours. RERR information are transmitted further if needed.

2.8 AODV-UU

AODV-UU is an implementation of the AODV (RFC3561) which run on GNU/Linux. The UU stand for Uppsala University a Swedish university.

This implementation works on all the 2.4 and 2.6 Linux kernels. A patch has been written to make it work with the kernel 2.6.39 which is used on the virtual testbed.

2.9 AODV-FUUREX

AODV-FUUREX is based on AODV-UU. This implementation add a mechanism of reputation.

2.9.1 Reputation mechanism

The reputation mechanism is based on the Ph.D. thesis of Francesco Oliviero [7] who proposed a reputation model called REFACING (RElationship-FAmiliarity-Confidence-INteGrity).
This reputation mechanism is based on three different reputation values.

1. **The local reputation** is managed by the node himself. Each node keeps the reputation of all its neighbours. To compute the value, the node checks that the packets are duly transmitted by its neighbours.

2. **The global reputation** is the reputation computed from the observation of the others nodes in the network.

3. **The current reputation** is the merge of the global and the local reputation which represents how much the node is trusted. This merging of reputation prevents reputation manipulation from the malicious node.

2.9.2 Local reputation computation

On each node a module called watchdog is launched. This module is responsible to check that the packet sent to the next-hop node are forwarded.
The Watchdog module is using the fact that wireless neighbour nodes are in radio range and that the data are emited all around the node.

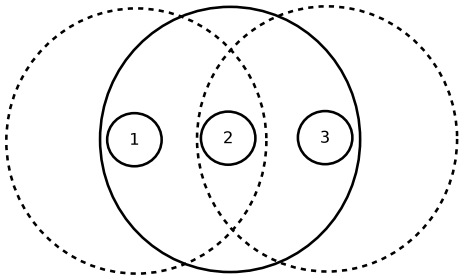
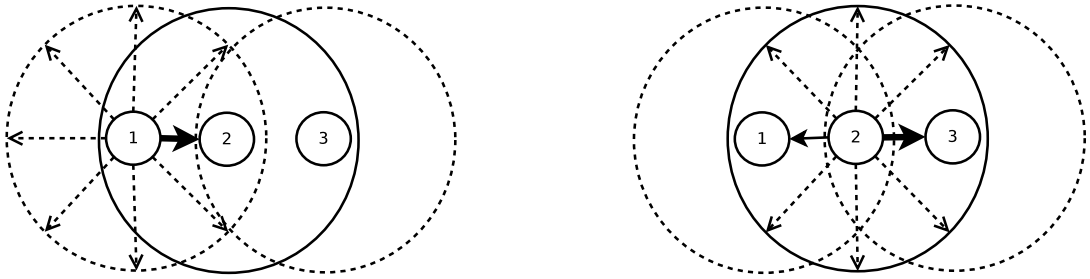


Figure 2.13: Node range

When Node 1 has sent a data to Node 2, watchdog keep the informations about the packet in a table for a fixed time. In the same time, Node 1 is sniffing all the packets which are in its radio range. When node 2 forwards the packet to Node 3, Node 1 received the same packet and can increase the local reputation of Node 2. If after a time, the packet is not forwarded Node 1 decreases the local reputation of Node 2.



(a) Node 1 sends data to Node 2 (b) Node 2 forward to Node 3 and Node 1 checks

Figure 2.14: Reputation security: Watchdog

2.9.3 Reputation table

To store the information of reputation on each node a list elements is created. In each element, the reputation information of a neighbour are stored with:

- The IP of the neighbour
- The last twenty computed values for the local reputation
- The last twenty computed values for the global reputation
- The number of packet forwarded to this neighbour
- The current reputation
- The average reputation
- The reputation variance
- The amount of reputation update

The last twenty values are used to compute a weighted moving average like explain in [7]

2.9.4 Global reputation dissemination

The dissemination of the global reputation is important and must be performed safely. Due to the broadcast nature of wireless network, the data could be used by the malicious node to take advantage from having the neighbours' opinion. To reduce the problem, the reputation propagated by the node is a merging of the local reputation and the reputation of the others. In this way, malicious node cannot detect which node started to propagate a bad reputation.
The global reputation of each neighbour is propagated by the RREQ messages in a special packet field added.
When a node receives a RREQ, it checks if there are neighbour reputations in the packet and update the global reputation with its local value.
The reputations of the neighbours of the node are added to the RREQ which is broadcasted to the neighbours.

2.9.5 Reputation in the route selection

In AODV, the route selected is the shortest path in hop-count. The length of the path is counted during the RREP process. In AODV-FUUREX, the route selection must be different and uses the reputation to select the most reliable route and not especially the shortest one.
The Hop-count in AODV-FUUREX is composed of the distance and correction factor which will increase the length of the path if the node's reputation is bad.
Another modification needed by AODV-FUUREX is to keep the different path values to compare all of them. In AODV, the first response is the shortest and the best route. In AODV-FUUREX, the shortest path could not be the best choice.

Chapter 3

Testing Methodology

In this chapter, the framework, the environments and all the configurations and scripts used to test AODV are presented.

3.1 Experimentation Framework

OMF (cOntrol and Management Framework) is a framework used to manage experimental testbed. OMF was developed at the beginning to work on ORBIT (a wireless testbed) and is now become more universal and work with a lot of technologies.
OMF is a set of tools which help in all the steps of experimentation. OMF is used to prepare the devices and deploy the data needed on each device.
During the experiment, OMF controls the devices and can take needed actions. Measurements are also made by OMF and can be post-processed when the experiment is finished.
The software is separated in multiple parts like on Figure 3.1

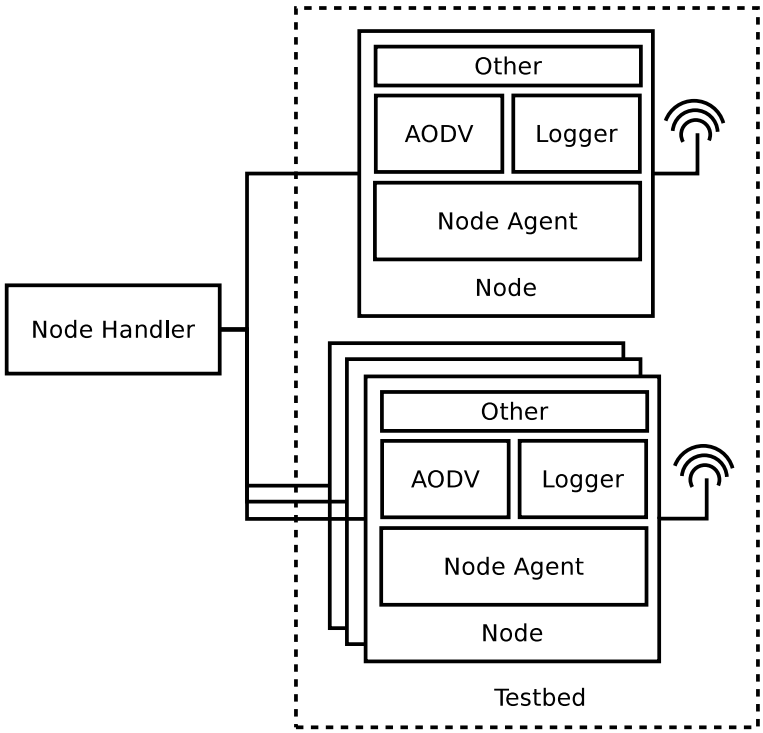


Figure 3.1: OMF components

- **The Node Handler** is installed on one machine to manage the experiments. It is this part which communicate with the nodes and sends the commands to the nodes. It's also the node handler which deploys the image disk and binaries needed to run experiments.
- **The Node Agent** is installed on each node. It listens and executes the command received from the Node Handler.

To make it easier a wrapper has been written by Giovanni di Stasi. With this layer, it is easier to make a topology and to manage the experiment running process.

3.2 Experimentation script

Like we have seen in the previous part, all the experiments use OMF to control the differents nodes. To make things easier, the experimental framework is helpful to reduce the complexity of the definition of the experiments. The experimentals scripts are written in Ruby like OMF. When we launch the experiment, we can specify differents parameters like:

- The topology
- The duration
- The data transfer speed and protocol type
- The routing protocol (AODV-UU or AODV-FUUREX)
- The activation/deactivation of the malicious nodes, the number and their activity scheme.
- The environment

All these elements must be specified before launching the experiment.

3.2.1 The topology

The topology is written in a file and gives all the information at the experimental script to configure each node according to its role. The roles are:

- Sender: Node which is used to send data in the network.
- Receiver: Node which receive all the data.
- Intermediate: All the nodes which are activated on the topology

The role of sender and of receiver must be specified at minimum one node. The experimental script allows more than one sender and receiver. The topology file contains also the informations about the bitrate of the wireless link. To complete the topology and create the different links between the nodes another file is used. This file contains the list of the links between 2 nodes. Only the nodes linked in this file can communicate together.
A link must exist between the sender and the receiver.
The nodes which are unreachable are considered as out of range node.

3.2.2 The duration

By default the duration of an experiment is fixed at 300 seconds. The duration of the experiment must be longer than 100 seconds which is the route refresh rate of AODV. After this duration all the routes are dropped and the RREQ procedure is restarted. To be sure that the network is well stabilized after the activation of the malicious node and the first route refreshed a delay of 100 seconds is added.
There are 2 phases in the experiment script.

- The first 1/5 the experiment is running without malicious nodes
- The last 4/5 of the experiment the malicious nodes are activated.

The separation in two parts enables to study the behaviour of the routing protocol without malicious nodes and let some time to check that everything goes well. It also enables AODV to create and choose the best route. In some case, without the first phase, the malicious nodes would be dropped directly and not considered during the route creation.

3.2.3 Data transfer speed and protocol type

To generate traffic, the experiment uses a traffic generator which is also configurable. The data transfer speed and the protocol type can be changed with two parameters of the experimental script.
It is important to be able to change the data transfer and to see the impact of the throughput on the network and on the CPU usage of each node.
The protocol type can also have an impact on the performance. A connection-oriented protocol like TCP will generate more traffic than a connectionless protocol like UDP which does not need acknowledgement.

3.2.4 The routing protocol

All the experiments are executed with the two implementations of AODV (AODV-UU and AODV-FUUREX). It is the best way to compare the two protocols and notice the differences.
To ease the switch, there is a parameter.

3.2.5 Malicious node management

The number of malicious nodes allowed for an experiment is N-2 (N=number of nodes in the topology). The sender and the receiver cannot be malicious. There are two differents types of nodes implemented to the experiment.

- Grey hole node: This type of malicious node filters the data and keep only the AODV traffic. All the other packets are dropped.
  To simulate this behaviour, the node uses iptable rules and allows only the AODV traffic to go through. This behaviour has the advantage that the node remains on a potential route because it answers all the RREQ and other AODV requests.
  This type of malicious node cannot be detected by AODV-UU which is not checking if the data traffic is transmitted.

- Black hole node: This malicious node drops packets of all types (AODV or data).
  This behaviour is obtained by netem kernel module which provides functionality to emulate delay, loss, duplication and re-ordering packet on a network. Two percentage values are provided to the modules.

  – The first value is the probability to drop a packet.
  – The second value increases the probability that the next packet is dropped too.

  With this configuration, black-hole can drop bursts and not only a single random packet. This type of malicious node can be detected by AODV-UU and AODV-FUUREX. If the number of dropped packet is too high, the node seems disconnected and the path is removed. Because packets are dropped randomly and the seed cannot be set, the results of the experiments are not reproducible and the comparison between two experiments is difficult.

3.2.6 Environment

- Virtual Testbed: The virtual testbed is composed of 5 virtual nodes which are linked by a virtual Ethernet bridge. This testbed is used to test the experiment script and check that everything goes well with AODV. The deployment of experiment on this environment is really faster than on Orbit and the usage is not limited in time. The biggest problem on this environment is the virtual link between the nodes. This link are more reliable than a wireless one's. There is no packet loss on this type of link.

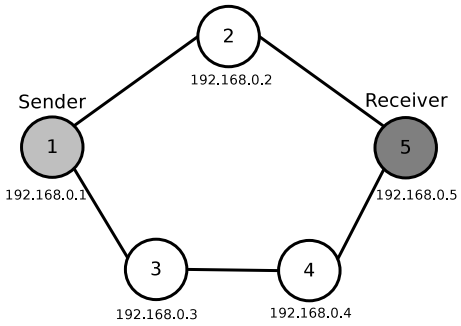


Figure 3.2: 5 nodes experiment topology

- Orbit: On orbit there are 400 nodes available. The reservation time is limited by slot of 2 hours. The experiment launched on Orbit are tested on the virtual testbed first.
  We have used the Virtual Testbed topology on Orbit to be able to compare the results.
  Another topology of 13 nodes has been also used to make some experiments. This topology uses more than one sender. A lot of issues were detected with this topology and will be discussed in section 4.3.
  On Orbit some nodes have other wireless chipset which are working differently. All the nodes must be checked before launching an experiment, to go quicker on Orbit and do not waste time, this topology was selected because it was tested for others experiments.
  The results with this topology were really useful to highlight problems but analyse was impossible.

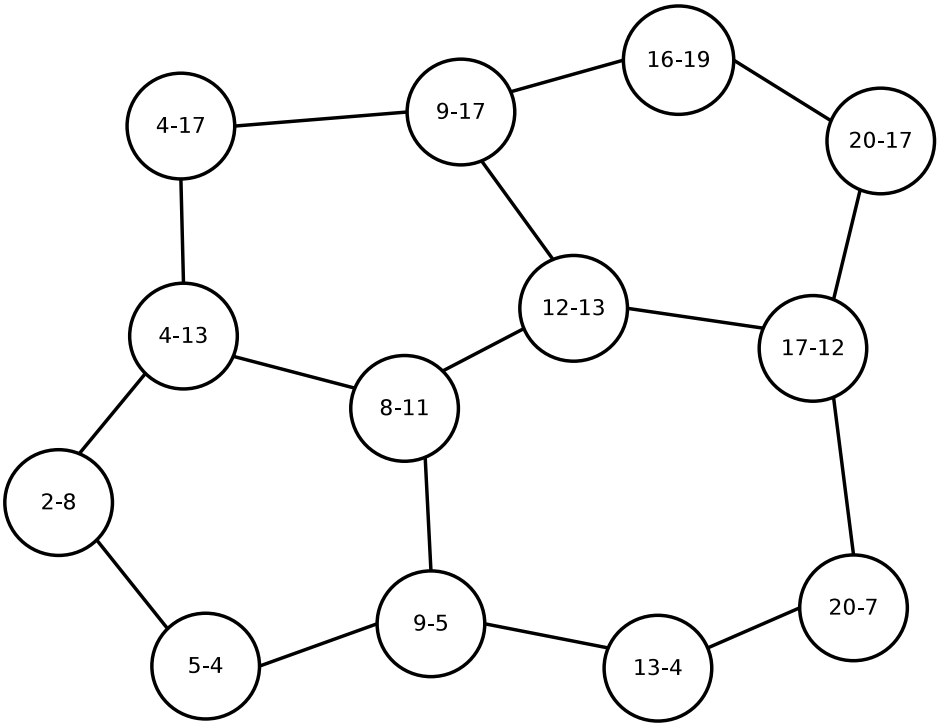


Figure 3.3: Orbit experiment topology

3.3 Binaries

To run the experiment, all the nodes must be configured with AODV. To make it, some files are uploaded on each node. All the binaries are on the main server and sent to the nodes at the initialization of the experiment. Some files are sent to the nodes regarding the parameters and roles in the experiment.

3.3.1 AODV-UU or AODV-FUUREX

The AODV implementation is the most important part of the experiment. This file is sent to all the nodes and contains the AODV program and the kernel module. There are different versions regarding the kernel version suited for the different kernel versions used on the virtual testbed and on Orbit.

3.3.2 Traffic Generator

Traffic is needed to check if everything is working on the network. D-ITG [1] is the traffic generator used for the experiment. This application is divided in different parts.

- A part of this application is installed on the sender(s) and generates the traffic on the network.
- Another part is installed on the receiver(s) and logs the traffic received in a file.
- A last part of this application is used to generate results from the log obtained during the experiment.

3.3.3 Logger

In addition to data obtained by the traffic generator which are only about network, a ruby script is installed on all the nodes to collect the CPU usage of AODV and the number of packets. This script logs all this data in a file. The data are by default sampled every 3 seconds. This value can be changed by a parameters when the script is launched.

3.4 Script in use

The experimental script is launched from a central server on which all the nodes are connected. All the binaries needed by the experimental script are on the server. When the script is launched, all the nodes are configured and the application is uploaded and launched.
When the script is launched there are different steps to install the nodes. The nodes are not connected together during the installation.

- Node loading: the binaries are uploaded on the nodes to be configured by the experiment script.
- Adding links between nodes: the topology is applied on the nodes and the links are configured.
- Configuring traffic generator: the traffic generator applications are installed on the sender and the receiver and the destination address is given to the sender(s).

[1] Distributed Internet Traffic Generator: `http://www.grid.unina.it/software/ITG/`

- Configuring AODV routing protocol: the kernel modules for AODV is installed and the AODV application is launched on all the nodes.
- Activation of the nodes and checking nodes: the network interface are activated and the script checks if all the applications are working. When this step is finished all the nodes are connected and the experiment is launched.
- A wait of 45 seconds is left to let AODV send HELLO message and to discover the neighbourhood.
- The logger and the traffic generator are launched.
- The first step of the experiment start (1/5 of the experiment time).
- The second step is launched (4/5 of the experiment time).
- The experiment is stopped. All the applications are stopped and the interfaces are closed down.

3.5 Post Processing

When the experiment is finished a script is launched to collect all the logs and interesting informations on the nodes.
With the collected data, an application provided with the traffic generator handle the data generated by the application.
The reputation log are handled by a Python script which computes time and generates a file for each neighbour of the node.
Gnuplot is used to plot automatically the graphs according to the data extracted before.

3.6 Validation

To validate the results, the experiments have been made more than 5 times on the virtual testbed. Due to the limited time on Orbit the experiments have been made 3 times maximum.

Chapter 4

Experimental Results

In this chapter, the results of the experiments are presented and discussed. Each presentation will be separated in two parts: a summary table and the discussion of the results including the reputation of the sender's neighbours computed by AODV-FUUREX.

4.1 Virtual Testbed Results

All the experiments presented in this section have been performed on the virtual testbed, with the 5-nodes topology shown on Figure 3.2.

Exp #0	AODV-FUUREX	AODV-UU
Time	300 seconds	
Bitrate	5 kbps	
Protocol	UDP	
Scenario	**No malicious Node**	
Throughput		
Delay		

4.1.1 Experiment #0 : without malicious node

The summary table is on page 29

- **The throughput** is the same for the two protocols. Without malicious node, the data transfer is never interrupted and the bitrate stays at 5 kilobits par seconds.

- **The delay** of AODV-FUUREX is little bigger than the delay of AODV-UU. This difference can be due to reputation computation.

- **Reputation**

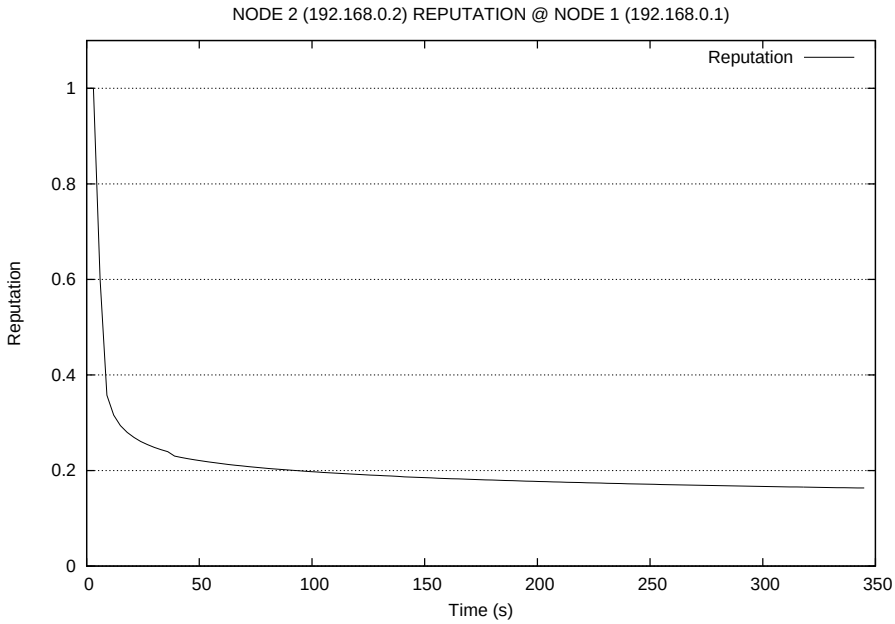


Figure 4.1: Experiment #O: Reputation node 2

Node 192.168.0.2 is on the shortest path. All data packets transit by this node to join the receiver. This graph shows a reputation computation problem. According to the throughput there is no packet loss but the reputation is decreasing nevertheless.

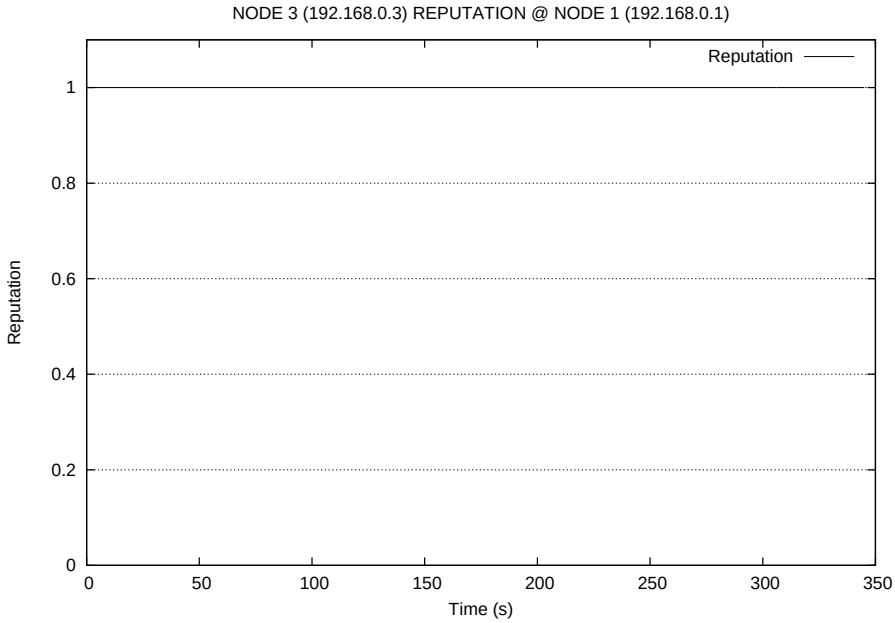


Figure 4.2: Experiment #O: Reputation node 3

Node 192.18.0.3 is never used by 192.168.0.1 to send data because it is not on the shortest path. There is no reputation computation and the value remains equal to 1 all experiment long.

- **Conclusion**
  This experiment confirms that AODV-FUUREX has the same behaviour that AODV-UU in a safe environment.

Exp. #1	AODV-FUUREX	AODV-UU
Time	300 seconds	
Bitrate	5 kbps	
Protocol	UDP	
Scenario	**Malicious node: (Grey hole dropping data packet @192.168.0.2 active from t=60s onwards)**	
Throughput		
Delay		

4.1.2 Experiment #1 : grey hole with low bitrate

The summary table is on page 32

- **The throughput** is better with AODV-FUUREX than with AODV-UU.
  AODV-FUUREX creates a new route at t=100s when the route are reset.
  AODV-UU cannot detect that the data are dropped by the malicious node. The traffic still flows through the shortest path and is dropped by the grey hole until the end of the experiment.

- **The delay** of AODV-FUUREX is little bigger than the delay of AODV-UU. This difference can be due to reputation computation.
  The gap in the graphs of AODV-FUUREX is due to packet loss during the period between the malicious node activation (t=60s) and the route reset(t=100s)

- **Reputation**

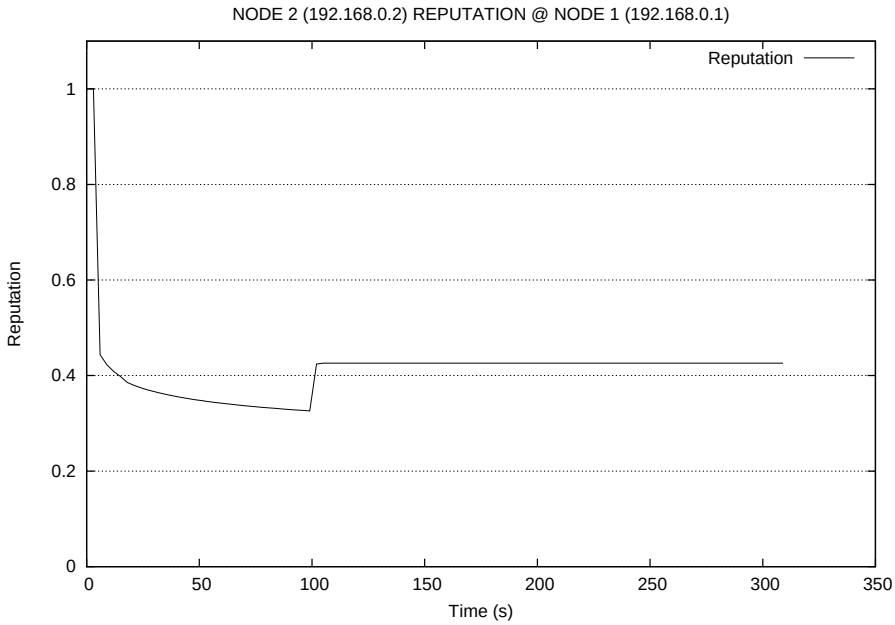


Figure 4.3: Experiment #1: Reputation node 2

As forwarder on the shortest path, this node is directly used when the experiment is launched. The reputation is decreasing directly and the activation of the malicious node at t=60s has no effect on the decrease of the reputation. When the route are reset and the node is no more used as forwarder, its reputation increases.

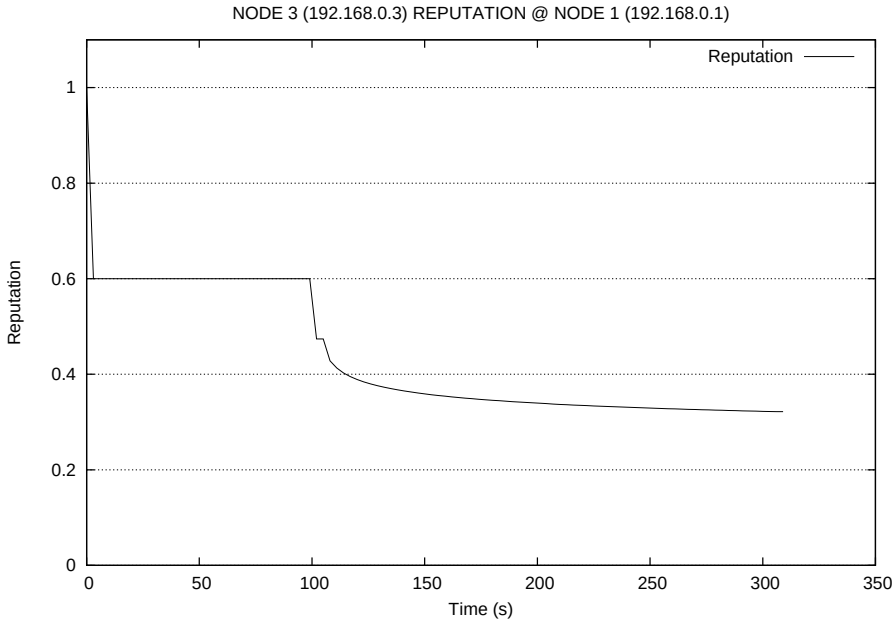


Figure 4.4: Experiment #1: Reputation node 3

This node is on the longest path and is not used at the beginning of the experiment. The activation of the malicious node at t=60s has no direct effect on its reputation. It is when the route are reset and that the node becomes the next-hop to join the destination that the reputation oddly starts to decrease. The reputation of this node remains higher than the malicious node until the end of the experiment.

- **Conclusion** In this experiment, there is a real problem in the reputation computation. The decrease of the reputation when nodes are forwarder cannot be explained by packet loss because the experiment is launched in a virtual testbed where there is no packet loss.

Exp. #2	AODV-FUUREX	AODV-UU
Time	300 seconds	
Bitrate	500 kbps	
Protocol	UDP	
Scenario	**Malicious node: (Grey hole dropping data packet @192.168.0.2 active from t=60s onwards)**	
Throughput		
Delay		

4.1.3 Experiment #2 : grey hole with high bitrate

The summary table is on page 35

- **The throughput** is not different with a higher bitrate. Like the previous experiment the traffic restarts after the route reset with AODV-FUUREX.
- **The delay** of AODV-FUUREX is one more time a little higher than the delay of AODV-UU.
- **Reputation**

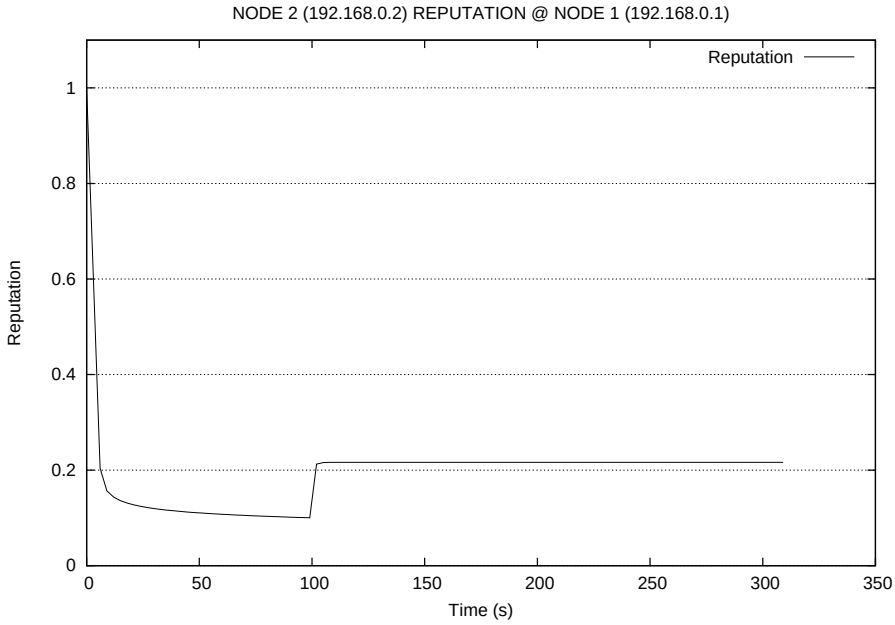


Figure 4.5: Experiment #2: Reputation node 2

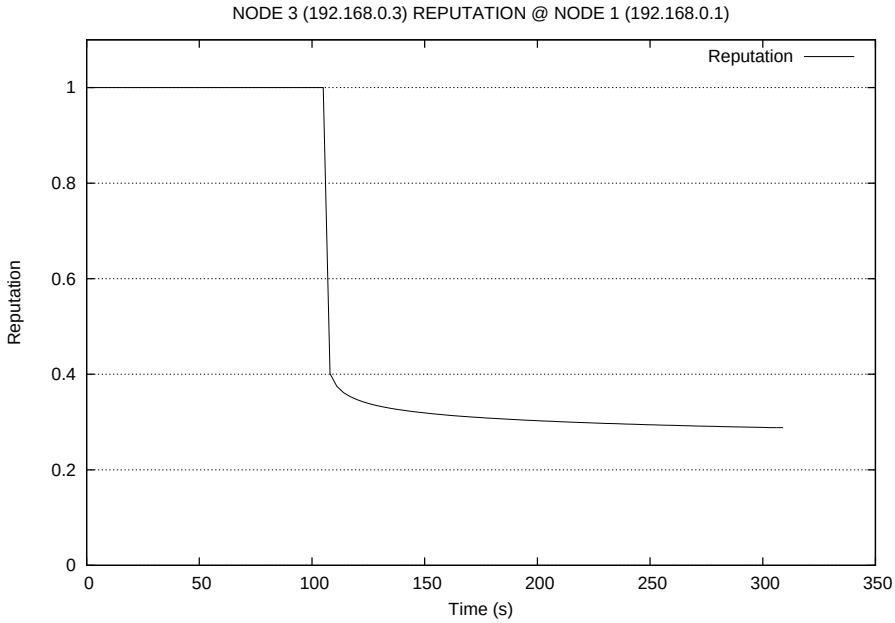


Figure 4.6: Experiment #2: Reputation node 3

- **Conclusion** As the bandwidth and computing are sufficient, a high bitrate has no effect on AODV-FUUREX and AODV-UU.

Exp. #3	AODV-FUUREX	AODV-UU
Time	300 seconds	
Bitrate	5 kbps	
Protocol	UDP	
Scenario	**Malicious node: (Black hole (50%/25%) @192.168.0.2 active from t=60s onwards)**	
Throughput		
Packet loss		

4.1.4 Experiment #3 : black hole

The summary table is on page 38
In this experiment it is harder to make comparaison between AODV-UU and AODV-FUUREX because the packets are dropped randomly.

- **The throughput** is interrupted by some packets drop.
- **The packet loss** graphs show only the number of data packets which are dropped. The AODV control packet dropped are not shown on this graph. During the experiment with AODV-UU more data packets are dropped than during the experiment with AODV-FUUREX.

- **Reputation**

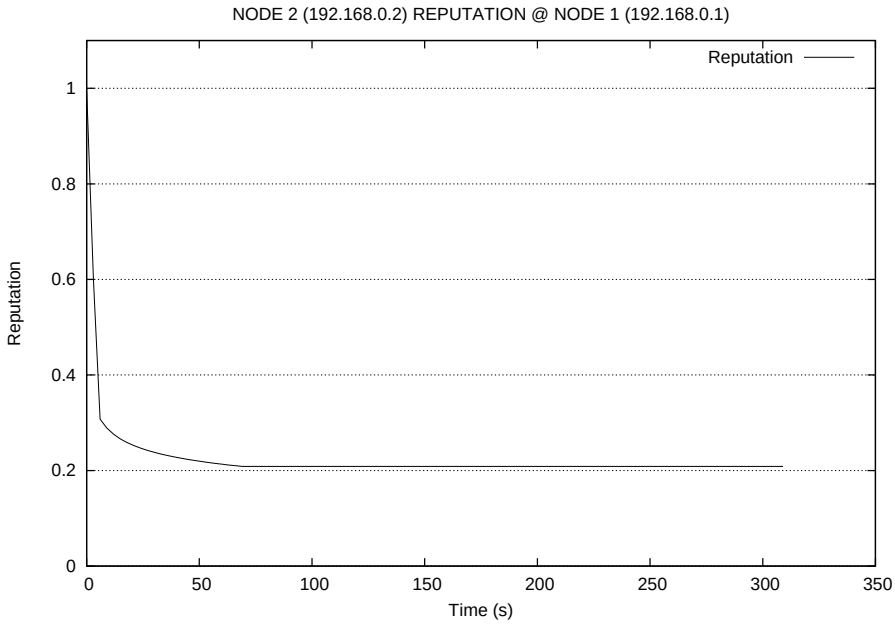


Figure 4.7: Experiment #3: Reputation node 2

Lies on other reputation graphs, the reputation of the node used at the beginning because on the shortest path is decreasing directly.

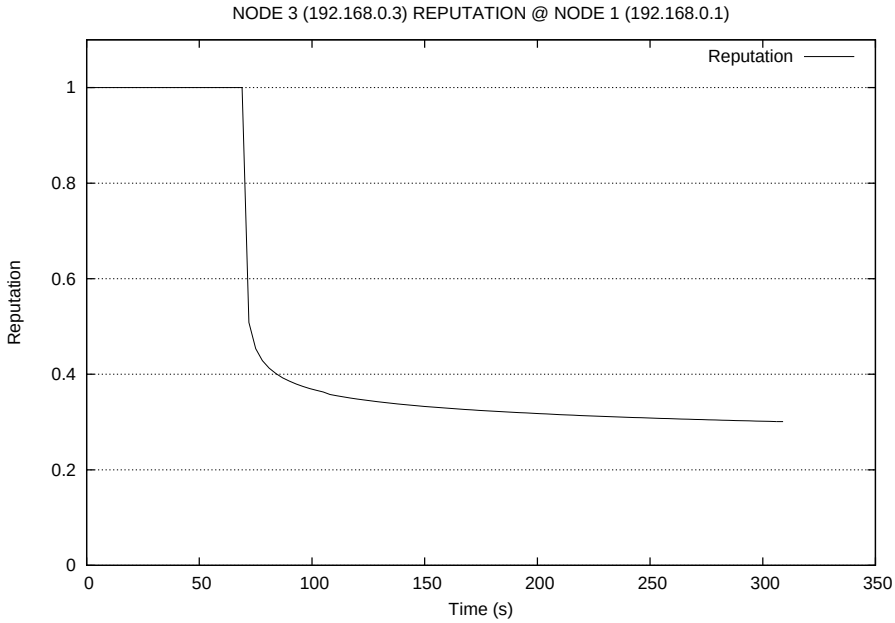


Figure 4.8: Experiment #3: Reputation node 3

On this graph, the reputation is decreasing from t=72s onwards.

- **Node 1 routing table**
  - For AODV-FUUREX

```
1 Destination  Next hop     HC St. Seqn Expire Flag Iface
2 192.168.0.3  192.168.0.3  1  VAL 1    1425         eth0
3 192.168.0.2  192.168.0.2  1  VAL 1    1425         eth0
4 192.168.0.5  192.168.0.2  2  VAL 2    3241         eth0
```

This routing table shows that at the beginning of the experiment the path to join the destination (192.168.0.5) go through 192.168.0.2

```
1 Destination  Next hop     HC St. Seqn Expire Flag Iface
2 192.168.0.3  192.168.0.3  1  VAL 1    1418         eth0
3 192.168.0.2  192.168.0.2  1  INV 2    12030        eth0
4 192.168.0.5  192.168.0.3  3  VAL 3    3306         eth0
```

At t=72s, the route to 192.168.0.2 becomes invalid and the next-hop to reach the destination becomes 192.168.0.3 which is on the longest path.

  - For AODV-UU

```
1 Destination  Next hop     HC St. Seqn Expire Flag Iface
2 192.168.0.5  192.168.0.2  2  VAL 1    2374         eth0
3 192.168.0.2  192.168.0.2  1  VAL 1    2374         eth0
4 192.168.0.3  192.168.0.3  1  VAL 1    2373         eth0
```

The first path selected by AODV-UU is the shortest one which goes through 192.168.0.2

```
1 Destination   Next hop      HC St. Seqn Expire Flag Iface
2 192.168.0.5   192.168.0.3   3  VAL 3    4318         eth0
3 192.168.0.2   192.168.0.2   1  INV 2    14988        eth0
4 192.168.0.3   192.168.0.3   1  VAL 1    1455         eth0
```

At t=81s, AODV-UU changes the path to destination like AODV-FUUREX.

- **Conclusion**
  In this experiment, AODV-UU and AODV-FUUREX have the same behaviour. Both of them change the next-hop some time after the malicious node activation.
  The malicious node drops AODV messages and the path to the node is invalidated. A RERR is generated and all the route which have the malicious node has next-hop are dropped and a new RREQ is directly sent to find a new route.
  This malicious behaviour is interpreted as an unreachable node.

Exp #4	AODV-FUUREX	AODV-UU
Time	300 seconds	
Bitrate	5 kbps	
Protocol	UDP	
Scenario	**Malicious node: (Black hole (25%/25%)@192.168.0.2 active from t=60s onwards)**	
Throughput		
Packet loss		

4.1.5 Experiment #4 : black hole

The summary table is on page 42
In this experiment it is harder to make comparaison between AODV-UU and AODV-FUUREX because the packets are dropped randomly.

- **The throughput** is interrupted by packet drop.

- **The packet loss** graphs shows the data packet dropped. One more time, the comparison between the two protocols is not possible. The packet are dropped randomly. In this experiment, there is more data packet dropped during the AODV-UU experiment. The path changes quicker with AODV-FUUREX because more control packets are dropped than with AODV-UU.

- **Reputation**

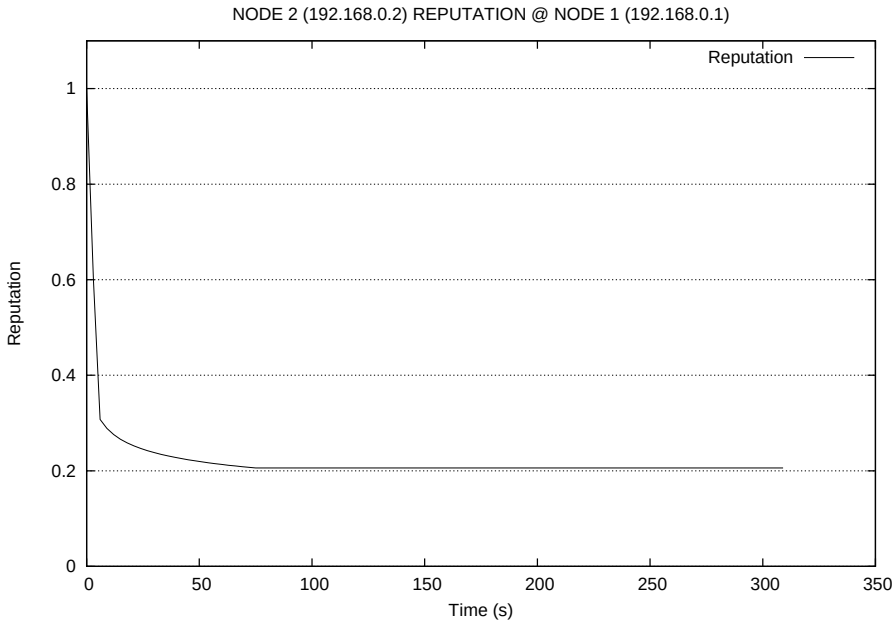


Figure 4.9: Experiment #4: Reputation node 2

For the node 192.168.0.2, the reputation is like in all the others experiments decreasing at the beginning.

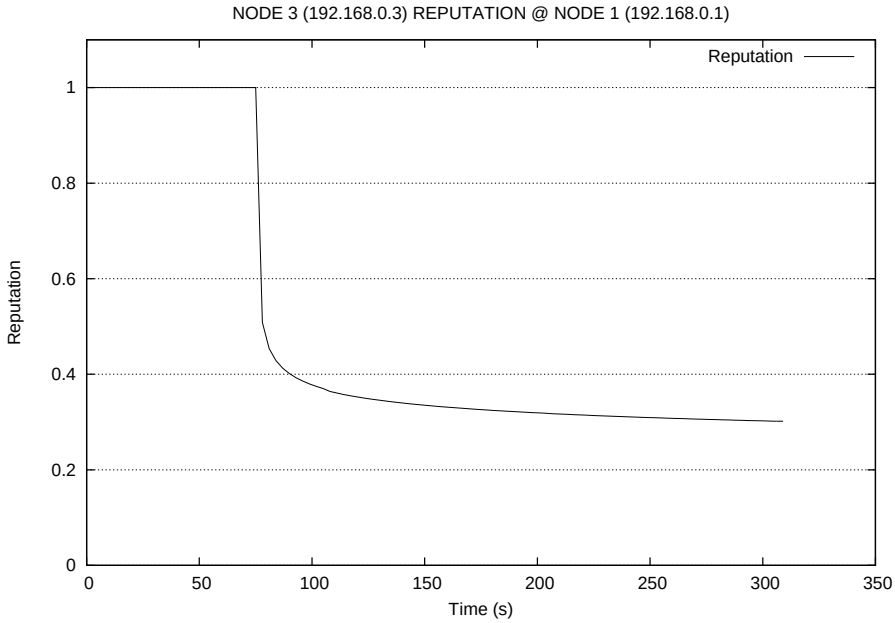


Figure 4.10: Experiment #4: Reputation node 3

It is at t=78s that the reputation starts decreasing. It is also at this time that the node is used as next-hop to join the destination.

- **Node 1 routing table**
  - For AODV-FUUREX

```
1 Destination  Next hop      HC St. Seqn Expire Flag Iface
2 192.168.0.3  192.168.0.3   1  VAL 1    1576         eth0
3 192.168.0.2  192.168.0.2   1  VAL 1    1576         eth0
4 192.168.0.5  192.168.0.2   2  VAL 2    3391         eth0
```

At the beginning of the experiment, it is the shortest path which is selected to the destination.

```
1 Destination  Next hop      HC St. Seqn Expire Flag Iface
2 192.168.0.3  192.168.0.3   1  VAL 1    1618         eth0
3 192.168.0.2  192.168.0.2   1  VAL 1    1618         eth0
4 192.168.0.5  192.168.0.3   3  VAL 3    3461         eth0
```

At t=76s, the routing table change and the node 192.168.0.3 becomes the next-hop to reach the destination.

  - For AODV-UU

```
1 Destination  Next hop      HC St. Seqn Expire Flag Iface
2 192.168.0.5  192.168.0.2   2  VAL 2    2224         eth0
3 192.168.0.2  192.168.0.2   1  VAL 1    1734         eth0
4 192.168.0.3  192.168.0.3   1  VAL 1    1734         eth0
```

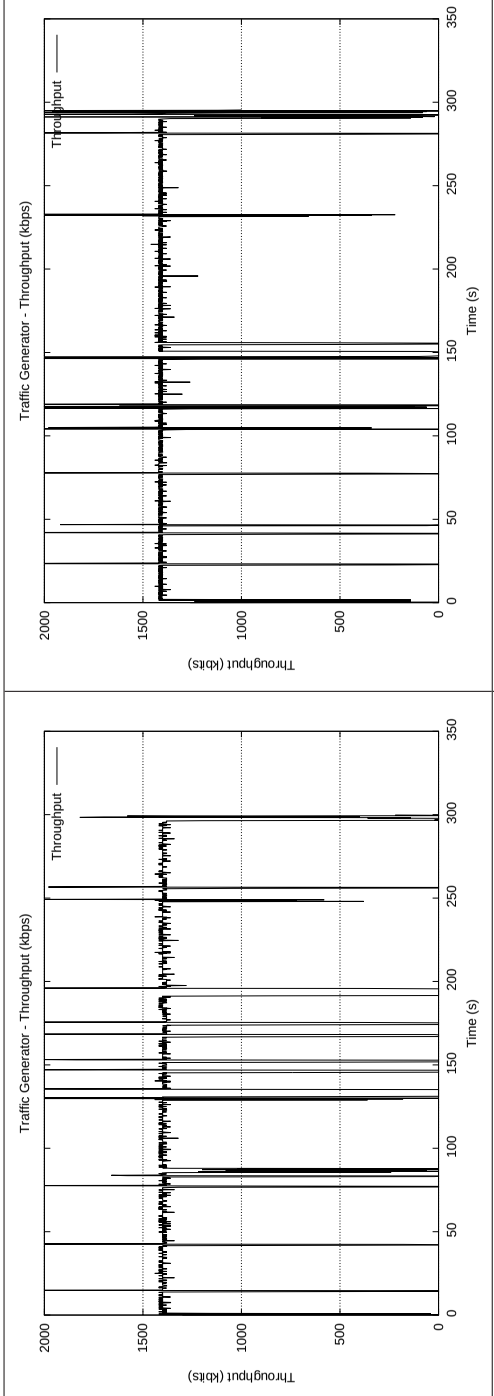
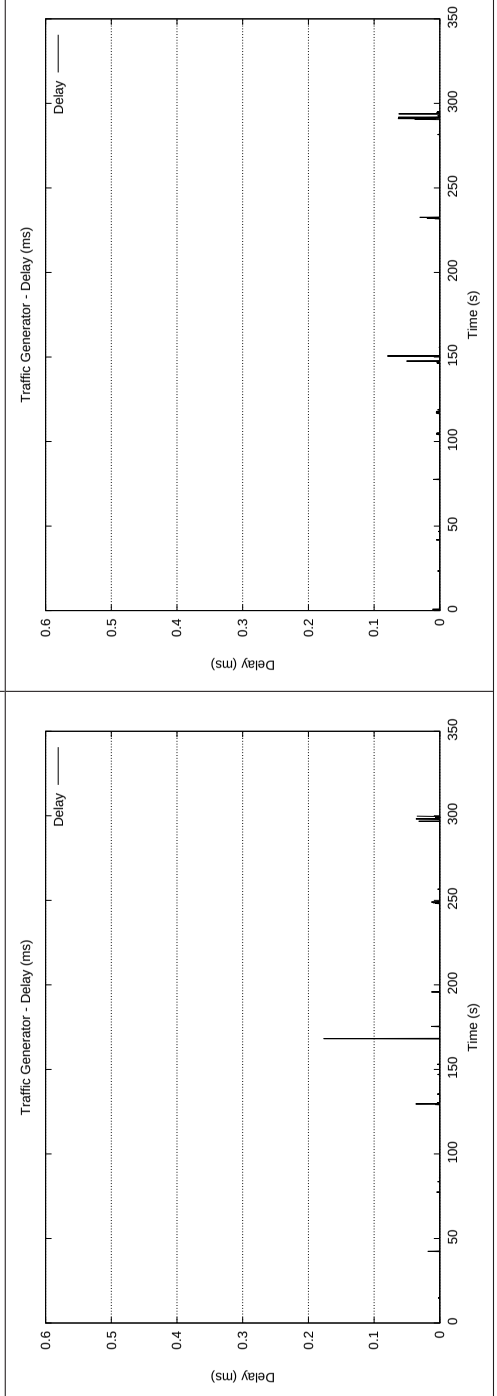
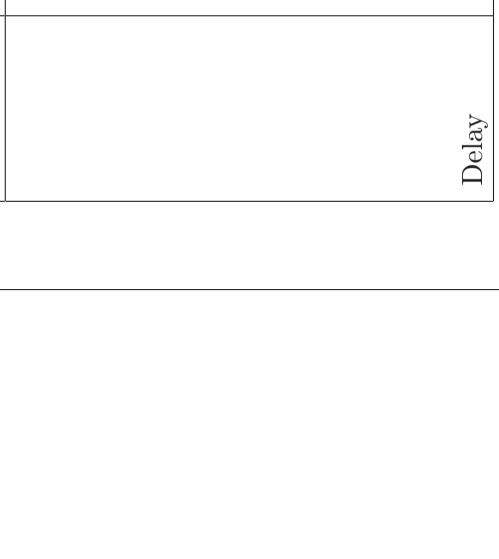
```
1 Destination   Next hop      HC St. Seqn Expire Flag Iface
2 192.168.0.5   192.168.0.3   3  VAL 7    3675          eth0
3 192.168.0.2   192.168.0.2   1  INV 2    14790         eth0
4 192.168.0.3   192.168.0.3   1  VAL 1    1463          eth0
```

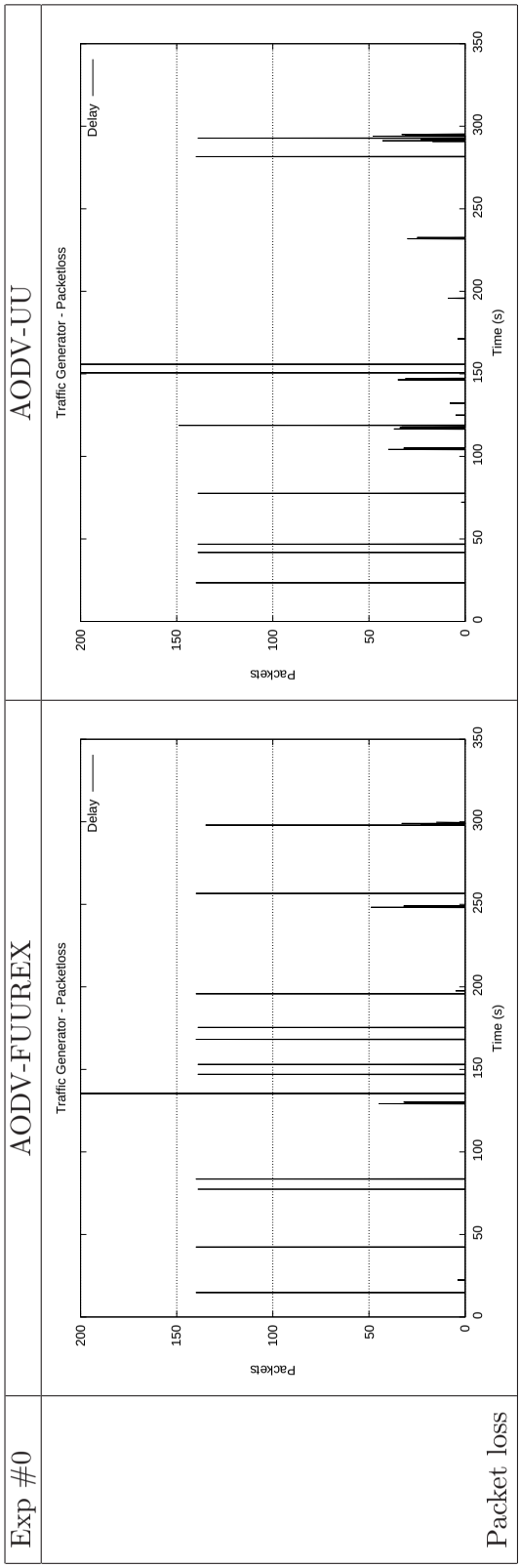
It's at t=117s that the route change and the next-hop to the destination becomes 192.168.0.3

- **Conclusion** There is more difference between AODV-UU and AODV-FUUREX than the other experiments with black hole. AODV-UU takes more time to change the route. This behaviour must be due to the type of packet dropped by the malicious node which drops less AODV control packet.

4.2 Orbit Testbed Results

In this section are results of the experiments made on Orbit with real node are presented.

Exp #0	AODV-FUUREX	AODV-UU
Time	300 seconds	
Bitrate	1400 kbps	
Protocol	UDP	
Scenario	**No malicious Node**	
Throughput		
Delay		


Exp #0
AODV-FUUREX
AODV-UU
Packet loss
Traffic Generator - Packetloss
Delay
Packets
Time (s)
0
50
100
150
200
250
300
350

4.2.1 Experiment #0 : without malicious node

The summary table is on page 47

- **The throughput** of this experiment shows clearly some random packet loss as expected in a real environment.

- **The delay** of AODV-FUUREX is lower than the delay on the virtual testbed. The highest peak is at 0.269 ms. This difference can be due to a better time synchronisation between the nodes and/or a better CPU computation which reduce the AODV-FUUREX computations.

- **The packet loss** corresponds to the throughput graph.

- **Reputation**

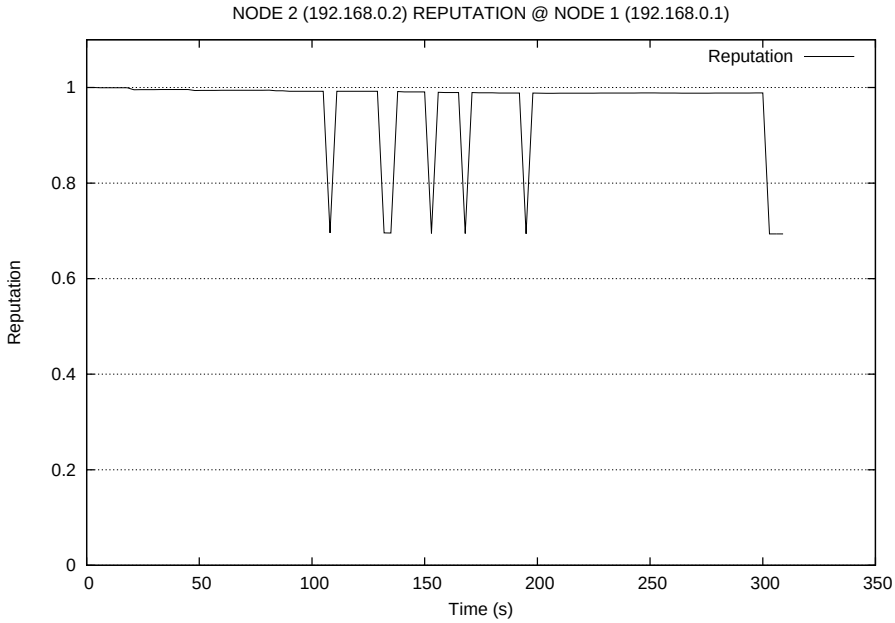


Figure 4.11: Experiment #0: Reputation node 2

The gap in the graph shows that the other path is sometime used. The routing information shows change between the two path for several seconds.

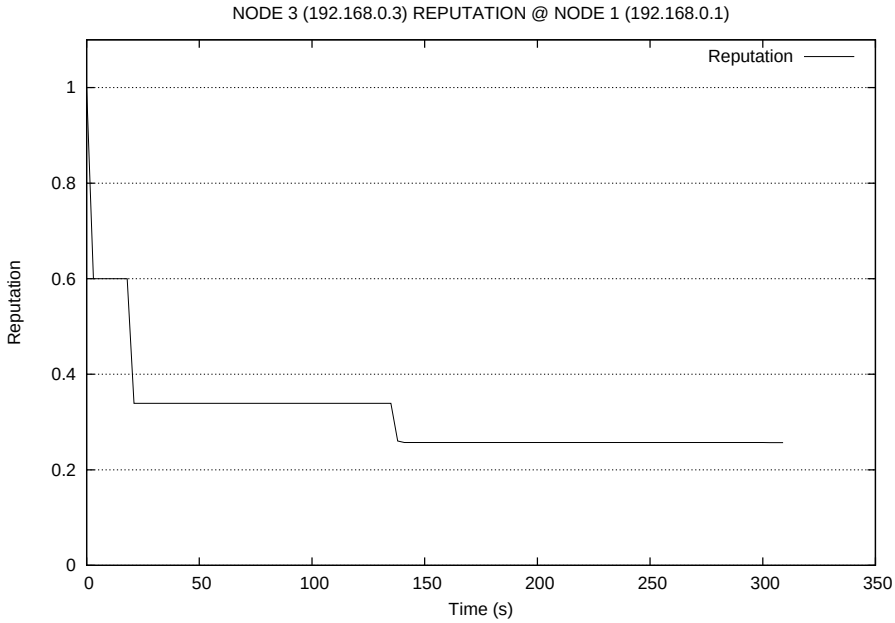


Figure 4.12: Experiment #0: Reputation node 3

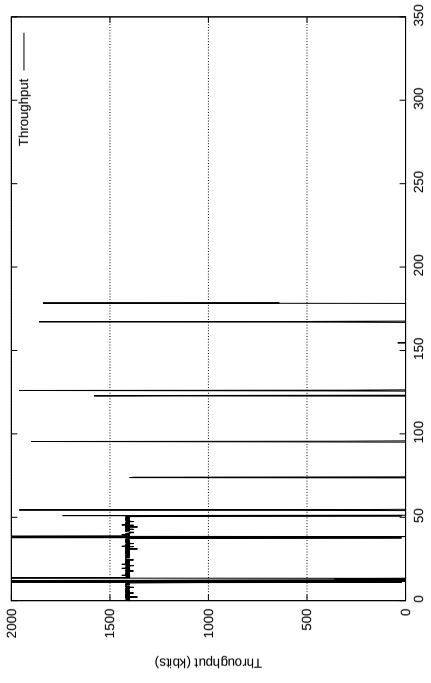
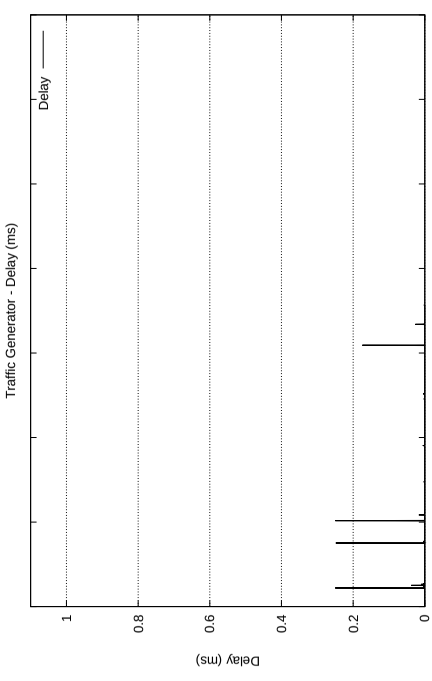
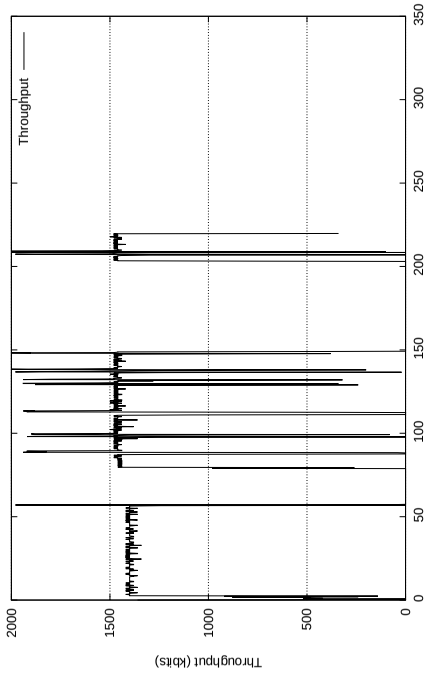
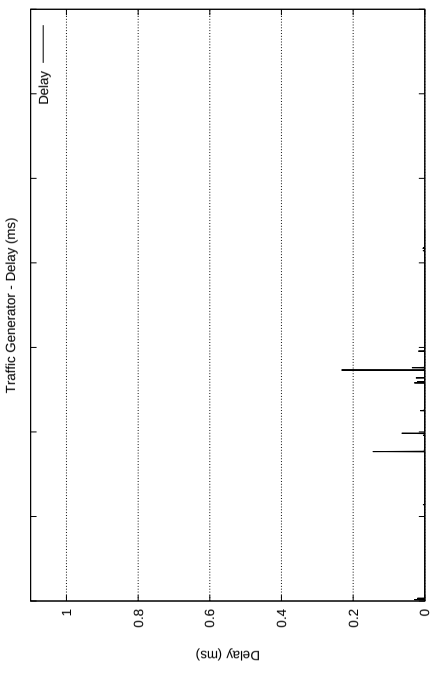
In this graphs, there is still a problem in the reputation computation. The node is not on the shortest path and is used only some seconds as we can see in the routing informations.

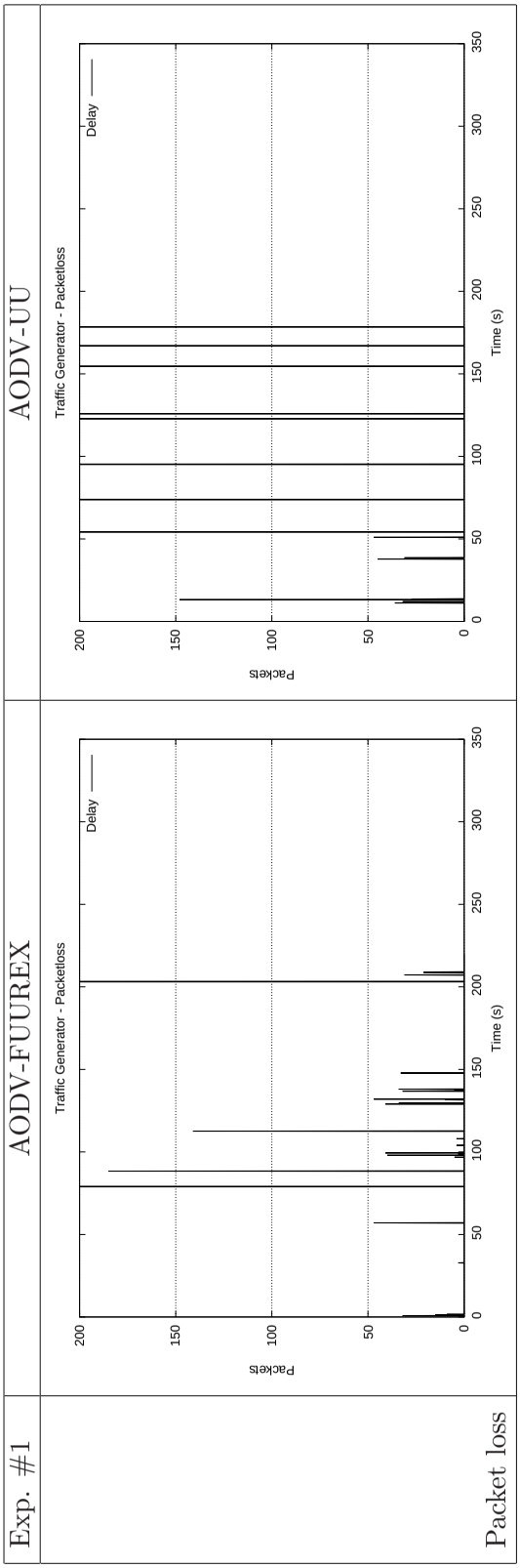
- **Node 1 routing table**

```
# Time: 06:56:06.858 IP: 192.168.0.1, seqno: 12
Destination  Next hop     HC St. Seqno Expire Flags Iface
192.168.0.2  192.168.0.2  1  VAL 1     1724         wlan0
192.168.0.3  192.168.0.3  1  VAL 1     1171         wlan0
192.168.0.5  192.168.0.2  2  VAL 12    2997         wlan0
# Time: 06:56:09.858 IP: 192.168.0.1, seqno: 13
Destination  Next hop     HC St. Seqno Expire Flags Iface
192.168.0.2  192.168.0.2  1  VAL 1     2265         wlan0
192.168.0.3  192.168.0.3  1  VAL 1     2271         wlan0
192.168.0.5  192.168.0.3  22 VAL 13    5274         wlan0
# Time: 06:56:12.859 IP: 192.168.0.1, seqno: 14
Destination  Next hop     HC St. Seqno Expire Flags Iface
192.168.0.2  192.168.0.2  1  VAL 1     1457         wlan0
192.168.0.3  192.168.0.3  1  VAL 1     1475         wlan0
192.168.0.5  192.168.0.3  22 INV 14    13323        wlan0
# Time: 06:56:15.859 IP: 192.168.0.1, seqno: 15
Destination  Next hop     HC St. Seqno Expire Flags Iface
192.168.0.2  192.168.0.2  1  VAL 1     1555         wlan0
192.168.0.3  192.168.0.3  1  VAL 1     1557         wlan0
192.168.0.5  192.168.0.2  11 VAL 14    3418         wlan0
```

This is an example of the next-hop changes during the experiment. The node 192.168.0.3 becomes the forwarder for several seconds. Most of the time, it is the node 192.168.0.2 which is on the shortest path which is used as forwarder to reach the destination.

- **Conclusion**
  The results of this experiment are really different that from the same experiment in the virtual environment. There is random packet loss which make change the route. The packet loss have a big influence on the reputation computation. On Orbit, it is the unused node which has a decreasing reputation at the beginning of the experiment.

Exp. #1	AODV-FUUREX	AODV-UU
Time	300 seconds	
Bitrate	1400 kbps	
Protocol	UDP	
Scenario	**Malicious node: (Grey hole dropping data packet @192.168.0.2 active from t=60s onwards)**	
Throughput		
Delay		


Exp. #1
AODV-FUUREX
AODV-UU
Traffic Generator - Packetloss
Delay
Packets
Time (s)
0
50
100
150
200
250
300
350
Packet loss

4.2.2 Experiment #1 : grey hole with high bitrate

The summary table is on page 53

- **The throughput** is in the AODV-FUUREX interrupted at t=60s like on the virtual testbed. The other route is taken before the route reset due to a invalidation of the route because of packet loss.
  For AODV-UU, the behaviour is like on the virtual testbed. There is some peak which come from a temporary route change due to packet loss.

- **The delay** is one more time lower than on the virtual testbed.

- **The packetloss** shows that there are some interruption in the transmission.
- **Reputation**

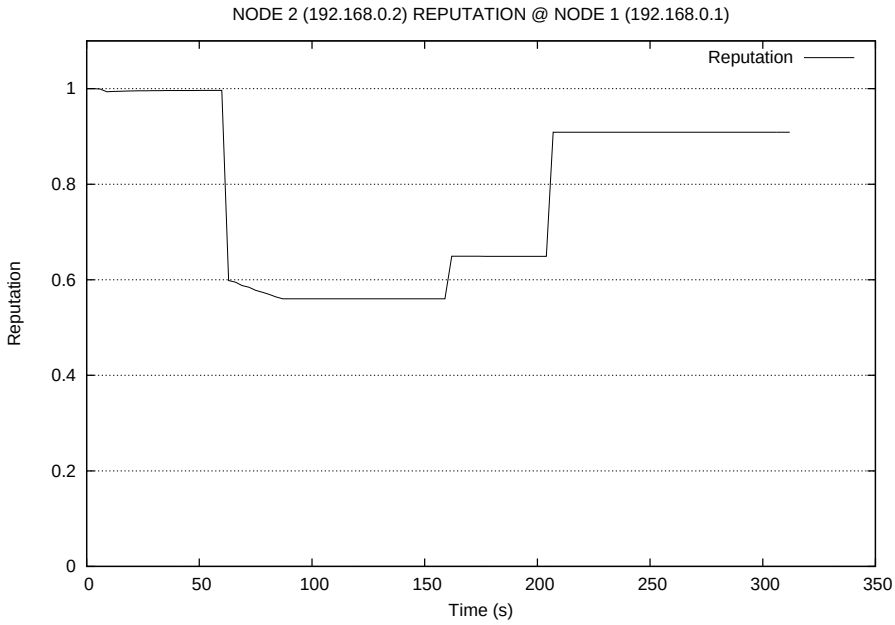


Figure 4.13: Experiment #1: Reputation node 2

This graph is the first where the activation of the malicious node is clearly visible. The reputation is decreasing at t=60s and not at the beginning of the experiment. The reputation increase later on is a problem because the malicious node is activated until the end of the experiment.

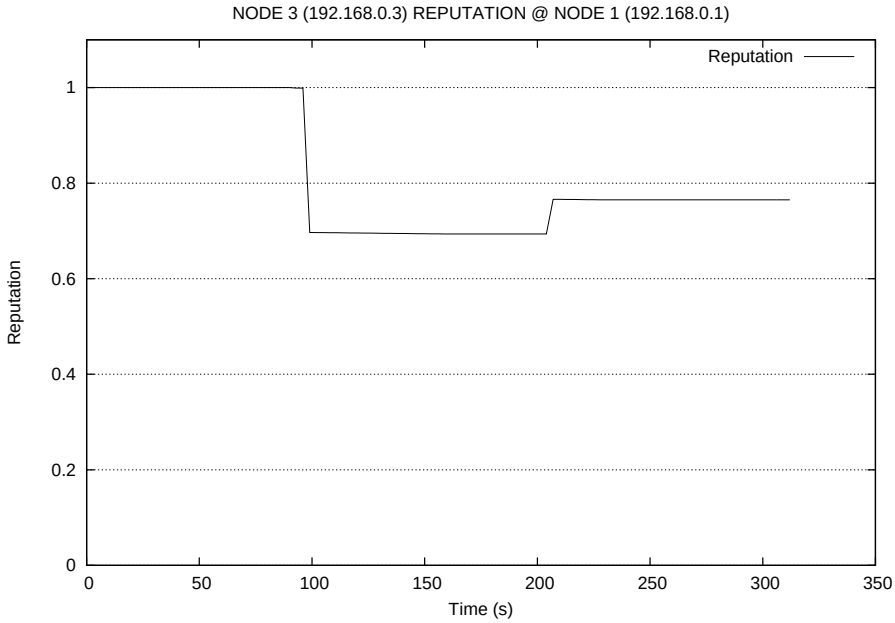


Figure 4.14: Experiment #1: Reputation node 3

In this graph, there is one more time some problem in the reputation computation. There is no real event which can explain why the reputation is decreasing. Node 192.168.0.3 is used as forwarder from t=85s to t=150s and from t=207s to t=222s.

- **Conclusion**
  The influence of the environment in the test on Orbit makes the analysis of the experiments really complex. The packet loss interacts in the route selection and is not always explainable.

4.2.3 Big Topology

The results of the test on the big topology were not good enough to plot graphs.
The sender was unable to reach the destination.

4.3 Problems detected

This part will present all the problems discovered during the experiments made with AODV-FUUREX. All the correction made are available on `http://git.infonet.fundp.ac.be/cgi-bin/gitweb.cgi/aodv-fuurex.git`

4.3.1 Kernel not supported

To use AODV-UU and AODV-FUUREX, a kernel module is needed. The current version of AODV-UU [1]and the provided version [2] of AODV-FUUREX works until the version 2.6.35 of the Linux kernel. The kernel used on the testbed is newer (version 2.6.39).
The AODV kernel module was not compilable. To fix it some kernel function call were changed. AODV-UU and AODV-FUUREX works now with Linux kernel 2.6.39.

4.3.2 Reputation decreasing until 0

Before using OMF on the virtual testbed, AODV-FUUREX was tested manually on the different nodes. To generate traffic, the ping (ICMP packets) was used to generate traffic and check the reputation computation.
During this test, the reputation was indefinitely decreasing until 0.
In the attempts to solve the problem, the traffic was recorded by tcpdump. According to this traffic capture, packets sent were not broadcast to all the nodes like it must be in wireless environment. Watchdog was not able to detect the transmission of the packets by the neighbours and was decreasing the reputation.
The problem was due to the virtual bridge used to connect the nodes. This bridge was configured to retain the mac address of the nodes to avoid the broadcast of packets.
To solve this problem the configurations of the bridge has been changed to avoid this behaviour.

- The Spanning Tree Protocol(STP) was deactivated.
- The route remain timer was put to 0.

After that all the packets were broadcasted and the reputation was computed for ICMP packets.

4.3.3 ICMP only

In the first tests on the virtual testbed with OMF the reputation was remaining on 1 for all the nodes and never changed.
Some test have been made without experiment script to find the problem. To generate some traffic without traffic generator the linux command "ping" was used. During the ping test the reputation was changing.
After some investigation the problem was detected in the implementation of the sniffer used by watchdog to listen to packets sent by the neighbour. The filter of the sniffer was configured to keep only ICMP traffic.
To solve the problem, the filter was changed to keep and transmit to watchdog all the data traffic. The AODV messages are not checked by Watchdog.

[1] `http://sourceforge.net/projects/aodvuu/`
[2] `http://git.infonet.fundp.ac.be/cgi-bin/gitweb.cgi/aodv-fuurex.git`

4.3.4 CPU peak

In the first experiments with a malicious node which provoke a route reset and a path change, the CPU usage of the sender was increasing until 100%.

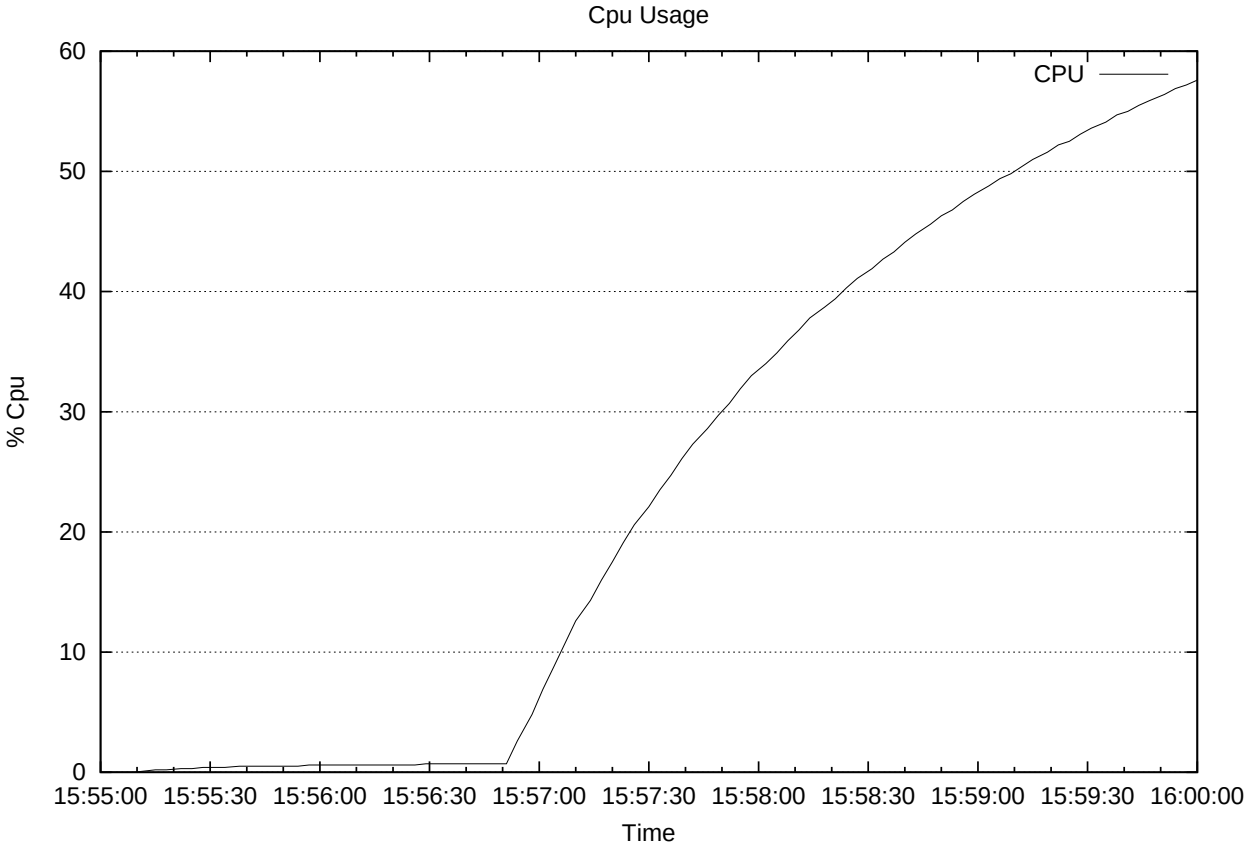


Figure 4.15: Cpu peak

As you can see on Figure 4.15, when the malicious node is activated at t=15:56:45 the CPU usage increases. This problem was finally due to a infinite loop in the cleaning of the queue of packet to check in watchdog for the old path. The last element of the list was pointing himself. After some modification in the implementation the problem was solve and the CPU usage lower.

4.3.5 Reputation computation error

In the reputation graphs shown here above the reputation of the node in use is directly decreasing without any reason. After investigation and debugging of AODV implementation some problems have been detected like:

- **Watchdog packet sort** A sniffer is used to provide the packets from the neighbours to Watchdog. The sniffer captures all the packets sent in the range of the node.
  Some packets must not be used to compute the reputation. There were some mistake in the filter used to sort the packets. These unfiltered packets were used by Watchdog and disturbing the reputation computation. The filter has been fixed.

- **Global reputation dissemination problem**
  The global reputation is normally sent to the other node in the option field of RREQ. After

a deep packet inspection of RREQ message the option field seems to be filled by AODV-FUUREX but the reputation value sent is always the same.
This problem is not fixed.

- **Reputation historic problem**
  Still in the attempt to fix the reputation computation problem another problem has been detected. When the global reputation is computed before being sent to the neighbours an average is made from the 20 last global reputation values.
  The global reputation history seems to be corrupted. All the values are equal and change at each computation of the reputation. The average computation is false.

```
1  WMA GLOBALE: value 1 = 0.714
2  WMA GLOBALE: value 2 = 0.714
3  WMA GLOBALE: value 3 = 0.714
4  WMA GLOBALE: value 4 = 0.714
5  WMA GLOBALE: value 5 = 0.714
6  WMA GLOBALE: value 6 = 0.714
7  WMA GLOBALE: value 7 = 0.714
8  WMA GLOBALE: value 8 = 0.714
9  WMA GLOBALE: value 9 = 0.714
10 WMA GLOBALE: value 10 = 0.714
11 WMA GLOBALE: value 11 = 0.714
12 WMA GLOBALE: value 12 = 0.714
13 WMA GLOBALE: value 13 = 0.714
14 WMA GLOBALE: value 14 = 0.714
15 WMA GLOBALE: value 15 = 0.714
16 WMA GLOBALE: value 16 = 0.714
17 WMA GLOBALE: value 17 = 0.714
18 WMA GLOBALE: value 18 = 0.714
19 WMA GLOBALE: value 19 = 0.714
20 WMA GLOBALE: value 20 = 0.714
```

  This problem is not fully fixed.

4.3.6 Conclusion

During all the experiments, problems occurs. Some of them are completely fixed, others will need more work. Here is a summary list.

Problem	Resolved	Reference
Kernel 2.6.39 support	✓	section 4.3.1
Testbed bridge	✓	section 4.3.2
Reputation for ICMP only	✓	section 4.3.3
CPU peak	✓	section 4.3.4
Watchdog filter	✓	section 4.3.5
Reputation dissemination	✗	section 4.3.5
Reputation history	✗	section 4.3.5

Chapter 5

Future works

The main part of this thesis was to create a working experiment script and test the available implementation of AODV-FUUREX. This chapter will present potential improvements.

5.1 Experimentation improvements

5.1.1 Scenario

The experimentation scenarios are limited. The experiment is divided in two parts, when the malicious node is activated or not. It could be interesting to be able to activate node by node or to deactivate the malicious behaviour of one node.

5.1.2 Topology

The test are made on 2 topologies with a limited number of node. Bigger topologies could be interesting to test the scalability of the network.
Some problems presented in the chapter 11 of [4] should be tested carefully.

5.1.3 Malicious Node

The current implemented malicious node are limited. More complex malicious nodes could be helpful to highlight the improvements of AODV.

5.1.4 Metrics

Results provided by the logs are limited to the main basics metrics. Some metrics could be added to improve the test. For example:

- The energy consumption of a node could be monitored. Mobile devices are often used in Ad-hoc network and works generally on battery. The reduction of energy consumption is one of the main objectives.
- The metrics provided by the traffic generator are only based on the data traffic. The AODV message statistics are not recorded. It could be interesting to know the impact of malicious nodes on the AODV traffic and to compute the overhead of AODV messages.

5.2 Implementation improvements

In the previous chapter, some implementation problems are explained and must be fixed. This section presents modifications which could improve AODV-FUUREX.

5.2.1 Linux kernel 3.0 supporting

The current version of the kernel module used by AODV-UU and AODV-FUUREX is not compilable with Kernel Linux 3.x. It could be interesting to use AODV with this kernel version which increasingly widespread.

5.2.2 Threshold to route reset

In all the experiment with malicious node, there is a delay between the detection of the malicious node and the route change. This delay could be reduced if the route reset was activated earlier.
A good move could be to provoke the route resetting when the reputation is under a fixed or computed value.
Another way to solve the problem without reset all the nodes could be to invalidate the route in the routing table to restart the process of RREQ without restarting all other routes.

5.3 Security weakness

This section presents some scenario which could be problematic. None of them have been tested.

5.3.1 Shielded malicious node

In this scenario, there are 2 malicious nodes working together. These nodes must be in border of the network.

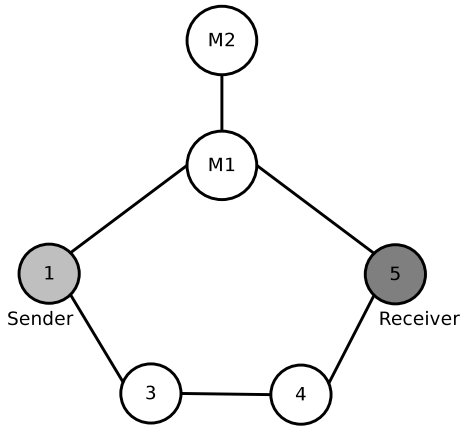


Figure 5.1: 2 nodes attacks

The first node is in the network (like M1 on the figure 5.1) and the second is only connected to M1 (like M2) With this configuration, M2 is completely isolated and the only neighbour in range is M1.
Two nodes are needed in this scenario to avoid problems with watchdog and a decreasing of the reputation. Node M1 must forward all the packet received to keep a good reputation. This

reputation will be also used to M2 which can inject forged packet with the reputation of M1. The forged packet can be used to make a black-hole and attract all the traffic, to disconnect node sending RERR or disturb the network. The node M1 is like a reputation shield.

5.3.2 Spoofing IP

In the current implementation of AODV-FUUREX, the reputation is only linked by the IP address as explained in section 2.9.3. If a node has a bad reputation, it can only change its IP address to have a better reputation.

Chapter 6

Conclusion

The goals of this thesis are to check the validity and test the performance of AODV-FUUREX which adds a secure layer on AODV-UU (an implementation of the RFC 3561).

With this secure layer all the nodes compute the reputation of their neighbours. The computed reputation is used in the route selection. If the reputation is bad, the node will changed the route if another exist.

To test this reputation, two types of malicious nodes were executed during the experiments. To check the validity and the utility of the reputation computation, tests were made on AODV-UU and AODV-FUUREX.
With the gray hole which is one of the implemented malicious behaviour. The results are clear. AODV-FUUREX detects the malicious node and changes the route to the destination. With AODV-UU, the route stays and all the packets are dropped after the activation of the malicious node.
The results with the black hole are more mitigated. AODV-UU and AODV-FUUREX discover the malicious node and change the route. When the drop rate of this malicious node is too important there are not enough "Hello message" transmitted and the route is invalidated. This leak of Hello message is interpreted as a disconnected node.

These tests have been made on two different environment which have also big impact on the results obtained.
On the virtual testbed, routes change only when there is a malicious node or when a node is disconnected. The throughput graphs are constant and there is no packet loss.
In this situation it is easy to see when the malicious node is activated and the computation of the reputation is normally easier.
Tests on the real testbed give other results. The graphs are more variable and there are more packet loss due to the radio interferences. It's harder to analyse.

During these tests problems occur. Problems are not easy to fix in network environment. The hardest thing is to identify the source of the problem. It can be in the experiment script, in the routing algorithm or in the test environment.
It becomes an advantage to have two different environments and two implementations of the routing algorithms. Change of environment or implementation can help to find the problem.
A lot of time has been taken to fix problems met during the experiments. The bugs in the environ-

ment configuration and in the experiment script are fixed but some remain in the implementation of AODV-FUUREX and will need more work to be completely fixed.
The main problem is in the reputation computation. In the results, the reputation of the node in use is decreasing without any reasons. A source of the problem has been detected but not completely fixed.

After all these tests and debugging, the current implementation of AODV-FUUREX sounds promising, the detection of grey hole works.
In the last chapter, some improvements are presented to reduce the reaction time when a malicious node is detected.
New tests with other malicious behaviours and bigger topologies must be done to continue the validation of AODV-FUUREX.

Bibliography

[1] Anu Bala. Investigation of blackhole attack on aodv in manet. http://ojs.academypublisher.com/index.php/jetwi/article/view/020296100/1837, 2010.

[2] Prashant Dewan Partha Dasgupta Amiya Bhattacharya. On using reputations in ad hoc networks to counter malicious nodes. http://www.cs.nmsu.edu/~amiya/pubs/icpads04.pdf, 2004.

[3] C. Perkins E. Belding-Royer S. Das. Rfc 3561: Ad hoc on-demand distance vector (aodv) routing. http://http://www.ietf.org/rfc/rfc3561.tx, 2003.

[4] Julien Van de Sype and Laurent Guillaume. A secure routing protocol implementation fuurex. No URL, 2010.

[5] Yih-Chun Hu Adrian Perrig David B. Johnson. Wormhole attacks in wireless networks. https://sparrow.ece.cmu.edu/group/pub/hu_perrig_johnson_wormhole.pdf, 2006.

[6] Kamarularifin Abd. Jalil Zaid Ahmad Jamalul-Lail Ab Manan. Mitigation of black hole attacks for aodv routing protocol. http://mimos.academia.edu/JamalullailAbManan/Papers/963336/Mitigation_of_Black_Hole_Attacks_for_AODV_Routing_Protocol, 2011.

[7] Francesco Olivero. On the effective exploitation of distributed information for cooperative network security and routing optimization. http://www.fedoa.unina.it/2063/1/Oliviero_Ingegneria_Informatica_Automatica.pdf, 2007.

[8] Mihail L. Sichitiu. Wireless mesh networks: Opportunities and challenges. http://www4.ncsu.edu/~mlsichit/Research/Publications/wwcChallenges.pdf, 2005.

[9] Priyanka Goyal Sahil Batra Ajit Singh. A literature review of security attack in mobile ad-hoc networks. http://www.ijcaonline.org/volume9/number12/pxc3871947.pdf, 2010.

[10] Umang Singh. Secure routing protocols in mobile ad-hoc networks - a survey and taxanomy. http://www.ijric.org/volumes/Vol7/Vol7No2.pdf, 2011.

[11] Ioannis G. Askoxylakis Nicolas Mechin George Perantinos George Vasilakis Apostolos Traganitis1. Usage scenarios and application requirements for wireless mesh networks. http://www.eu-mesh.eu/files/publications/INDUSTRY-01.pdf, 2009.

[12] Yih-Chun H U and Adrian Perrig. A survey of secure wireless ad hoc routing. http://www.cs.jhu.edu/~cs647/class-papers/Security/AdHocSurvey.pdf, 2004.

[13] Laurent Guillaume Julien van de Sype Laurent Schumacher Giovanni Di Stasi Roberto Canonico. Adding reputation extensions to aodv-uu. http://wpage.unina.it/rcanonic/papers/2010.SCVT/P00_036_Camera_Ready.pdf, 2010.

[14] Nital Mistry Devesh C Jinwala Mukesh Zaveri. Improving aodv protocol against blackhole attacks. `www.iaeng.org/publication/IMECS2010/IMECS2010_pp1034-1039.pdf`, 2010.